JPRS L/10391

16 March 1982

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 4/82)

FBIS FOREIGN BROADCAST INFORMATION SERVICE

FOR OFFICIAL USE ONLY

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040037-2

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000500040037-2

JPRS L/10391

16 March 1982

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 4/82)

CONTENTS

HARDWARE

Data Control Medium Modules for Automation of Physical Optics Research 1

Basic Applications of Bubble Memory for Microcomputers 8

Flat Magnetic Domain Memory Modules With Advanced Level of Integration 18

Interfacing of Digital Strain Gage Bridges With 'Elektronika-60' Microcomputer 26

Signal Panel for Automated Control System 29

Alphanumeric Combined Keyboard for Readout of Characters in 16-Bit Position Code 30

Miniature Vibrationproof Multipoint Electrical Connector 33

Automation of Testing and Adjustment of Hybrid Integrated Circuit Resistors With Computer Control 35

Accessory Module for Input of Initiative Signals 40

System for Input of Analog Signals Into M-6000 Computer 42

Shch68400 and Shch68000K Printers 45

Graphic Data Input Unit 48

Data Display Unit With Dynamic Indication 50

GRAPHIC DISPLAY UNITS

'Grafit' Graphic Display 52

Character-Graphic Information Display 58

Miniature Information Display Unit Utilizing IMG-1 Display 61

Color Character-Graphic Display 63

SOFTWARE

Abstracts From Journal 'PROGRAMMING', November-December 1981 65

Introduction to Theory of Analysis and Parallel Operation of Computer Programs During Translation 69

Using Parallel Calculations Model To Investigate Structure of Oriented Graphs 73

Programming PL/1 Computer System 80

Software of Type M5000 Computer Complex and Description of Its Elements 86

System Approach to Automated Design of Microprocessor Computers ... 92

Automation of Development of Programs for Data Recorders 94

APPLICATIONS

Application of Modern Microprocessor Technology in Problem-Oriented Systems 98

Automation of Information Processes in Integrated Automated Industrial Management Systems 112

ORGANIZATIONS AND PERSONALITIES

Moscow University's Computer Center 129

Fiftieth Birthday of Kamil' Akhmetovich Valiyev 136

HARDWARE

UDC 681.327.8

DATA CONTROL MEDIUM MODULES FOR AUTOMATION OF PHYSICAL OPTICS RESEARCH

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 10, Oct 81 pp 9-11

[Article by V.G. Tsyvinskiy, candidate of technical sciences, and G.G. Bondarovich and N.A. Sokolova, engineers]

[Text] The job of creating automated systems for scientific research is one of the most important. The experience of developing systems for automating technological processes has pointed out the need of developing general-purpose complexes based on digital control and computing equipment making it possible by means of configuration to construct various automation systems. Existing automation complexes of the ASVT-M [modular computer hardware system], KAMAK [CAMAC] and KTS LIUS [local data control system hardware complex] types reflect in their structure the ideology of a third-generation computer, which involves the concentration of the "intellect" and information in a single center--a processor with an on-line memory.

The development of microprocessor engineering has provided an opportunity for a new approach to solving automation problems, by dispersing the "intellect" over the system and by bringing it as close as possible to the points of the pickup and output of information. One possible solution is the creation of data control media (IUS's), which must lend themselves easily to configuration; adjustment of the medium for a specific object (the apparatus) can be called "loading" the test bed into the IUS. Interaction of the experimenter with the apparatus and medium is accomplished by means of an experimenter's automated work place (console) furnished with a screen type of display and an external memory.

It is possible to design control systems based on IUS's by placing certain limitations on their structure. For example, the structure of an IUS can be in the form of a "tree." Let us note that the majority of experimental apparatus has the same structure: It contains a basic subsystem (the top of the tree) connected with the output for the main "product" of the apparatus, and a number of support subsystems interconnected hierarchically. An IUS with a tree structure can in a certain information sense duplicate the structure of the apparatus, whereby subsystems of the IUS interact with respective subsystems of the object, performing the pickup of experimental data and subsystem control (fig 1).

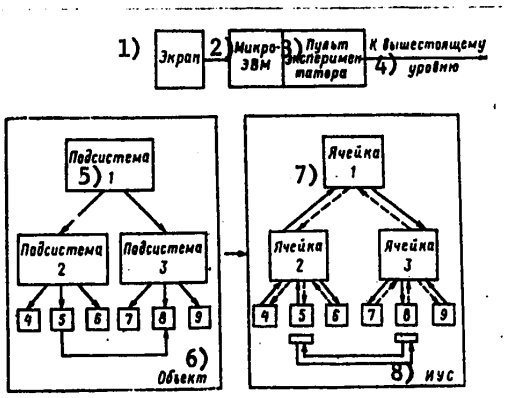


Figure 1. Structure of Data Control Medium

Key:

1. Screen
2. Microcomputer
3. Experimenter's console
4. To superior level
5. Subsystem 1
6. Object
7. Cell 1
8. Data control medium (IUS)

Thus, each subsystem is able to be connected to one superior subsystem and several inferior subsystems can be connected to it. Obviously, with this IUS structure each superior branch point of the tree can act as a redundant facility for any inferior one. Below is considered a variant with redundancy of a level of one. This makes it possible easily to share resources between two neighboring levels and at the same time to ensure a sufficiently high degree of reliability (stability). It is possible to demonstrate that the probability of failure of the system depends on the mean branching ratio of the tree, the total number of subsystems and the total number of IUS "clusters," Q , whereby this probability is less the higher number Q is. Even for the minimum value of Q (Q = 3) the probability of failure is reduced five- to sixfold, and for Q = 50 , by more than 1.5 orders.

Structurally a subsystem consists of a plug-in module (system of buses, connectors and installation points) which can include a microprocessor (MP), memory units (an on-line semipermanent memory (PPZU) and ROM), an adapter for connection between the processor and an upper level and modules for interfacing with the object, and in which points are provided for the connection of adapters for connection between the processor and inferior subsystems (fig 2). All components of a subsystem are unified on the basis of a single parallel interface.

A subsystem must swap data with the object (apparatus) through a small (maximum of 10) number of information channels and accordingly the number of object interface modules is not great. This is an important fact governing the flexibility of the system. Lists of modules and tables of routines in subsystems are thereby easily

visible, they are simple to put together and easy to change. The physically dispersed nature of the software determining the system's "intellect" thus improves not only the stability but also the metastability of the system.

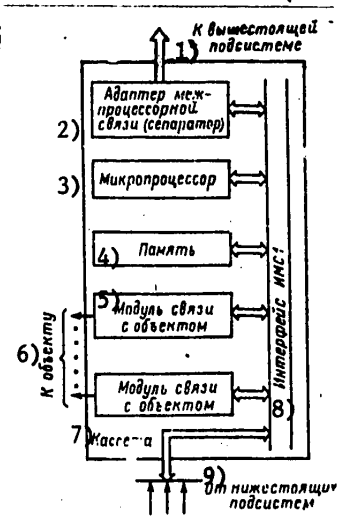


Figure 2. Subsystem (Cell) of Data Control Medium

Key:

1. To superior subsystem
2. Processor interface adapter (separator)
3. Microprocessor
4. Memory
5. Object interface module
6. To object
7. Plug-in board
8. IMS1 interface
9. From inferior subsystems

A subsystem serves as a building block for a combination of any configuration of the tree type. When a direct link is required between subsystems (without access to superior subsystems) it can be organized by means of duplex registers connected to individual subsystems (in fig 1 such a link is shown between subsystems 5 and 8). Thus, a structure of the oriented graph type can be organized. However, it is impossible to organize redundancy by means of duplex registers; therefore, with respect to redundancy the structure represents a tree.

Let us discuss the redundancy of the "intellectual" (microprocessor) module in greater detail. First, redundancy makes obligatory designing of the interface according to the parallel principle (an interface of the IMS-1 type, "Common Line"). Second, the microprocessor interface adapter must make possible connection (and accordingly disconnection from) with the superior subsystem of like lines of the redundant subsystem's interface. Thereby a single "transparent" interface is formed under the control of the microprocessor module of the superior subsystem. The execution of routines is slowed somewhat (the operation of subsystems becomes serial), but it is not halted. Obviously, the address fields of the on-line memory and of object interface modules of subsystems adjacent vertically, in order to avoid lack of uniqueness, must not meet, but must be divided among levels

adjacent vertically. The control routines of a subsystem must be allocated to the ROM, which is part of the microprocessor module. With this solution these routines (identical for each subsystem) can occupy the same absolute region of the memory, i.e., they are not taken into account in dividing memory resources among subsystems.

Unlike computing media, in data control media the redistribution of tasks among subsystems in the operating process does not take place; such a redistribution is observed only when changing the structure of the automated object (the apparatus) and also when a cell (subsystem) is made redundant.

Data control media can be used to automate experiments of various classes. Furthermore, the structure of object interface modules can be changed. In particular, for the purpose of automating physical optics studies a set of object interface modules has been developed which is designed for working within the structure of IUS subsystems designed on the basis of a type 580 IK 80 microprocessor with an output to an IMS-1 interface.

The following basic principles were observed in designing a set of modules for these purposes.

The program exchange algorithm is the most important algorithm for interaction between modules and the microprocessor.

For modules triggered by the research object either the exchange of information in the memory direct access (PDP) mode is provided, or the output of data to a RAM of the magazine type (OZUm) connected directly to the module's output. In the program exchange mode the MP assigns the operating mode for these modules (number of measurements, size of memory array allotted).

Disruption of the normal operating mode results in an interrupt enabling signal's (ZPR) being sent to the module. In units in which a ZPR can be caused for more than one reason, the output of a state word determining the reason for the interrupt or its place, to the interface in the program exchange mode, is provided for.

The proposed set includes the following modules.

1. Module ATsP1, consisting of a multiplexer, a level fixer (FU) and a digital-balance analog-digital converter (ATsP) which are electrically not interconnected, which makes it possible to put together various structures from them. Upon the "Output" instruction the module perceives the output data as the address of the multiplexer's channel, and upon the "Receive" instruction outputs the measurement result to the interface.

Also provided in the module is the serial interrogation of channels with a frequency determined by the ATsP's conversion time or by an external oscillator. In this mode information is output from the ATsP's output similary to program exchange upon the ATsP's initiative without outputing the address of the information receiver's unit (in the magazine-type RAM).

The parameters of the module are as follows: number of channels--16 with the possibility of expanding to 32; number of ATsP bits--8; interrogation time for a single channel--10 μs; input voltage from 0 to +5 V; error--0.5 percent.

2. The voltage switching (KM) module is similar to the ATsP1 multiplexer and can be used independently or together with the ATsP1 module for constructing a 256- (512-) channel multiplexer with 2-stage switching.

3. An OZUm [magazine-type RAM] module designed for the serial input, storage and serial output of a data array. This module has an output to the interface and an independent input for the write-in of information from peripheral units (e.g., ATsP1).

Interaction with the multiprocessor is carried out in the program exchange mode. When the memory is filled the module stops receiving and sends an interrupt enabling signal to the interface. Service requests from a peripheral unit and the MP are attended to in order of arrival. Parameters: capacity--512 bytes (with possibility of expansion to 1024); write-in time--1.5 μs.

4. A clock (GVI) designed for forming a sequence of clock pulses arriving with a specific frequency. This module represents a controlled frequency divider (frequency of an internal or external oscillator). The division factor is set by means of the interface in the program exchange mode in the form of a 2-byte pulse: The basic division factor in binary code and the additional (determining additional division of the frequency by 10, 100 or 1000) in a 3-bit parallel unit-counting code.

Parameters: basic division factor--1 to 100 with intervals of 1; additional--10, 100 or 1000; frequency of internal oscillator--10 MHz $\pm$ 10 percent with instability not greater than 0.01 percent.

The set described above is supplemented by the following modules:

5. An interface controller module enabling step-by-step representation of the control routine with an indication of the state of interface lines. This module is designed for connection to any connector of the interface and for reproduction from the control console of all signals put out by the microprocessor (duplicates the microprocessor).

6. Module ATsP2--an 8-bit analog-digital converter of the conveyer type with a conversion time of 1 μs. The ATsP is started through an external signal. Information is output to the magazine-type RAM or to the interface in the memory direct access mode.

7. A calibration signal source module which forms through external instructions two calibration potentials (approximately zero and approximately the upper limit of the dynamic range) and a test signal which increases stepwise at a rate determined by an external frequency. The test signal has four slope values which are also chosen through external signals.

8. A normalizing amplifier module consisting of four d.c. amplifiers with variable gain (10, 100 and 1000) and the ability to operate in the current-to-voltage conversion (PTN) mode. The operating mode (gain or PTN) is set via the interface in the program exchange mode.

9. A multipoint analog memory module (AZU)--a 32-point analog memory designed as a deviation-controlled slave system with the correction of errors introduced by the stray capacitance of switching elements. The rate of switching of memory elements to the input (output) of a module is determined by an external frequency and the number of readings stored is determined by the number of clock pulses arriving.

10. A control module--a special-purpose unit for controlling the subsystem for the investigation of rapidly occurring processes (BU).

Two possible structures for designing a data control medium (IUS) cell are suggested on the basis of the modules described (fig 3).

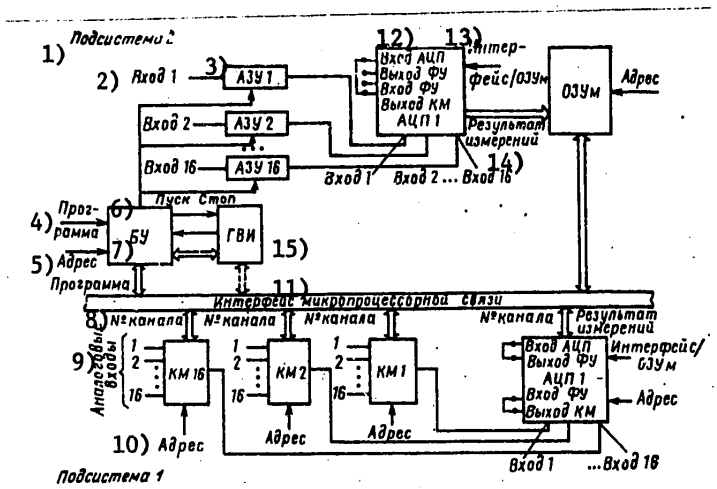


Figure 3. Structure of Data Control Medium Subsystems for Automation of Physical Optics Investigations

Key:

1. Subsystem 2
2. Input 1
3. AZU 1
4. Program
5. Address
6. Start, stop
7. BU
8. Number of channel
9. Analog inputs
10. Address
11. Microprocessor interface
12. ATsP input, FU output, FU input, KM output, ATsP1
13. Interface/magazine-type RAM
14. Measurement result
15. Clock

The first is designed for program interrogation of 256 sensors with relatively slowly changing signals. The normalizing amplifiers are not shown in the figure. In this structure identical addresses are assigned to commutators and analog-digital converters. The four lower-order bits read out in data lines are the channel address in multiplexers of the first stage (the KM module) and the four higher-order bits in the second stage (the ATsP1 multiplexer). An individual sensor is connected to the ATsP's input according to the address read out, of the required channel in data lines, accompanied by the "Output" instruction. The analog-digital converter is started by means of internal circuits. Upon the termination of conversion data are read out from its output upon the "Receive" instruction.

The second structure is designed for measuring the instantaneous values of voltage pulses arriving simultaneously through 128 channels. The required number of readings (instantaneous values) in the input signal is stored by means of analog memories (AZU's). Stored values are read out by the analog-digital converter (ATsP) at the conversion pace. Data are read out from the ATsP to the magazine-type RAM for subsequent program readout by the microprocessor. Control of the system is accomplished by means of a control module which is assigned its operating mode by the mulitprocessor (number of channels to be interrogated, pace and number of required readings). Calibration of the measuring section is provided for in the cell.

Both structures can be united in a single data control medium cell.

8831
CSO: 1863/43

UDC 681.327.6.004.122

BASIC APPLICATIONS OF BUBBLE MEMORY FOR MICROCOMPUTERS

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 10, Oct 81 pp 11-14

[Article by B.N. Naumov, USSR Academy of Sciences corresponding member, V.K. Rayev, candidate of technical sciences, and G.I. Markarov, engineer]

[Text] Bubble memories from the moment of their origin were intended to replace external electromechanical memories, primarily magnetic disks. One of the main obstacles for this replacement is the relatively high cost of bubble memories. Nevertheless, such important advantages of a bubble memory as independence of power and low power consumption, high information density and speed of response, the lack of rotating mechanical parts and the ability to increase the capacity of the memory block by block, make profitable the use of bubble memories in many general-purpose and special-purpose units. Of special interest is the use of a bubble memory in units based on microprocessor sets and in microcomputers, where requirements for memory capacity are often limited to figures on the order of 10 Mbits.

A number of foreign firms have already begun the series production of bubble integrated microassemblies (DIM's) and memories based on them, designed as a rule for use as the external memory (VP) of a microcomputer. The key parameters of series DIM's suitable for use together with large-scale integrated circuit microprocessor sets are presented in table 1.

Three mains trends are known in the use of bubble memories for microcomputers [1]: for constructing an external memory, for expanding an on-line memory (OP) and for designing a control memory (UP). The use of a bubble memory controlled by a microprocessor is promising for organizing dynamic redistribution and associative search in relative data bases, ordering information, editing texts, etc.

The goal of this article is a preliminary analysis of the effectiveness of using bubble memories for microcomputers in the three trends listed.

Floppy disk memories (NGMD's) and cassette memories (KNML's) are used as external memories in the majority of microcomputers. The key characteristics of modern NGMD's, KNML's, bubble memories and charge-coupled memories are presented in table 2 [2], from which it can be concluded that bubble memories considerably surpass in all their technical data electromechanical floppy disk and cassette memories.

Table 1.

Firm, country	Type of DIM	Structural organization	Capacity in bits	Transfer rate in Mbits/s	Mean access time in ms	Power requirement in W	Number of terminals	Overall dimensions in mm
Texas Instruments, USA	T1B0203	P-S[1]; 144X641	92K	0.05	4	0.7	14	-
	T1B0303	P-S, 224X1137	254K	0.1	7.3	0.9	18(20)[2]	30X30X10
	T1B0250	P-S, 274X1025	256K	0.085	5.6	1.2	24	33X35X9.5
	T1B0500	P-S, 274X2049	512K	0.085	11.2	1.2	24	33X35X9.5
	T1B1000	P-S, 2(274X X2049)	1024K	0.17	11.2	1.2	24	33X35X9.5
Rockwell International, USA	RBM256	P-S, 260X1025	260K	0.15	4	0.82	18	30X30X10
	RBM411	P-S, 572X2052	1024K	0.15	8	-	-	-
National Semiconductors, USA	NBM2256	P-S, 256X1024	256K	0.1	7	0.75	16	30X28X9
	NBM2201	P-S, 583X2048	1024K	0.1	11	0.9	16	30X28X9
Fujitsu, Japan	FBM32DA	P-S[1]; 138X587	81K	0.05	4.5	0.5	18	31X28X10
	FBM310B	P	74K	0.1	7.4	0.5	18	31X28X10
	FBM43DA	P-S, 265X1033	270K	0.1	6	0.67	20	31X28X11
Intel Magnetics, USA	7110	P-S, 256X4096	1024K	0.1	41	0.6	20	43X42X11
	7112	P-S, 256X4096	1024K	0.2	20	0.2	20	43X42X11
Plessey, England	PB064	P[3]	64K	-	-	-	12	-
Siemens, FRG	RBM256[4]	P-S, 260X1025	260K	0.15	4	0.82	18	30X30X10
Motorola, USA	"	"	"	"	"	"	"	"

[1]With closed interface register. [2]Two types with 18 and 20 terminals. [3]Open serial register.
[4]Licensed by Rockwell International (USA). P-S = parallel-serial; P = parallel.

A charge-coupled memory is better in terms of speed of response characteristics than a bubble memory, but such an important disadvantage of a charge-coupled memory as its power dependence makes it practically unsuited for constructing an external memory.

Because of the ability to increase the memory's capacity block by block, characteristic of bubble memories, their unit cost almost does not depend on information capacity. The approximate relationship between unit cost and information capacity for bubble memories, floppy disk memories and cassette memories (curves 1, 2 and 3, respectively) is presented in fig 1. Although the unit cost characterizes memories only in the static mode (storage mode), even with respect to this indicator, as is obvious from fig 1, a bubble memory with a capacity of 3 to 4 Mbits surpasses a floppy disk memory.

Table 2.

Memory characteristics	Type of memory			
	Bubble	Charge-coupled	Cassette	Floppy disk
Access time	4-10 ms*	100 μs	40 s	300 ms
Data transfer rate	800K bits/s	2 Mbits/s	1CK bits/s	250K bits/s
Frequency of occurrence of errors	10^{-12}	10^{-10}	10^{-7}	10^{-8}
Mean time between failures in hours	40,000	7000	3000	4500
Unit power requirement in μW/bit	10	15	8	35
Unit cost in cents/bit	0.1-0.2	0.2-0.25	0.07	0.15
Power independence	Yes	No	Yes	Yes

*Reduction of this time to 0.1 to 1 ms in the future involves the use of current accessing [10] and the employment of a 2-stage hierarchy in the storage structure of a bubble chip.

Memories are characterized more fully by the price of productivity, taking into account speed of response parameters [3]. For external memories designed for use in a specific system it is determined as follows: $P_v = C_v(T_v + N_b/v)$, where C_v is the unit cost, T_v is the access time, N_b is the size of a transferred block of data and v is the data transfer rate. Minimization of the price of productivity can serve as a criterion for selecting the type of external memory. From the condition $P_v^{\text{SMD [bubble]}} < P_v^{\text{NGMD}}$ it is possible to obtain the region of

external memory capacities in which bubble memories are more efficient than floppy disk memories. With the values of speed of response characteristics presented in table 2 and dependences of unit cost on information capacity presented in fig 1 the upper limit of this region is about 20 Mbits.

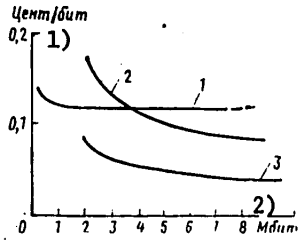


Figure 1. Dependence of Unit Cost on Information Capacity for Bubble, Floppy Disk and Cassette Memories

Key:
1. Cents/bit
2. Mbits

Since an external memory capacity of not greater than 10 Mbits is basically needed for modern microcomputers, bubble memories are more effective in this application than are floppy disk. Approximately the same results are gotten when comparing bubble memories with cassette memories, whereby the use of the latter is quite limited because of their relatively slow speed of response and low reliability. Thus, at the present time it is wisest to design a microcomputer external memory based on a bubble memory. A considerable advantage of a bubble memory is the ability to place the external memory on the same circuit board with the microprocessor and at the same time to create a microcomputer on a single board.

A criterion of no small importance in selecting an external memory is the simplicity of the controller. Bubble memory and floppy disk controllers are identical for the most part, which is due chiefly to the similarity of the control functions performed by them. The main principle for the implementation of a programmable controller for a floppy disk memory by Rockwell International (USA) designed for use in a microcomputer [4] consists in the use of an "external data block" placed in the microcomputer's on-line memory and designed for controlling the organization of the write-in format and for forming tag codes. This approach not only makes possible sufficient flexibility in controlling the format but also makes it possible to reduce the number of controller registers needed for this and at the same time to minimize the structure of the floppy disk memory controller.

The use of a bubble memory in contradiction to the use of a floppy disk memory makes possible the parallel (byte by byte) access to information, which eliminates the need to implement in the controller functions of parallel-serial conversion when writing information in and serial-parallel when reading it out. In addition, a bubble memory easily makes possible a start-stop operating mode. This, first, makes it possible to realize the write-in and readout of information at a variable

speed, which in the majority of cases eliminates buffering of data. Second, it assumes rapid response of the system and consequently makes possible addressing without the use of tag codes. The latter fact considerably simplifies and sometimes even completely eliminates from the controller's structure a format control block and an address and data comparison block and simplifies the instruction decoding and control block. On the whole, the controller of a bubble memory can become approximately twofold smaller than a floppy disk memory controller in terms of hardware input.

The bubble memory controller made by Texas Instruments (USA), whose structural diagram is shown in fig 2, is a general-purpose programmable controller designed for use in a microcomputer [5]; it stores information on the instantaneous position of data in storage registers, makes possible a start-stop operating mode and generates signals for controlling the functions of a bubble memory. Universality is achieved by means of software loading in the internal registers of the controller of individual parameters determining the structure of the chip and of the entire bubble memory and the size of a data block which can be transferred. Thereby an on-line change in the contents of internal registers makes it possible for the controller to work with a bubble memory consisting of chips of various structures and capacities and to accomplish the transfer of information in blocks of various lengths and in several blocks in succession. Through instructions arriving from the microprocessor the controller organizes the execution of the functional operations required for accessing one or more blocks of data from the bubble memory.

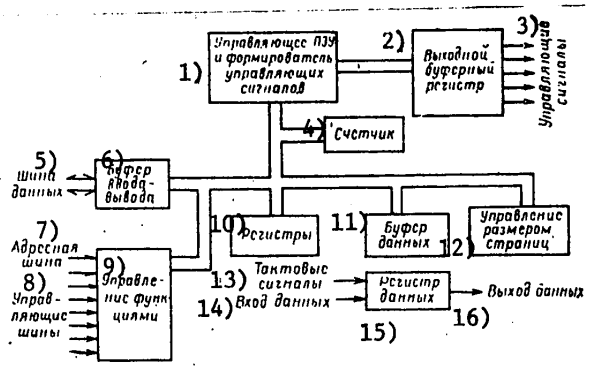


Figure 2. Structural Diagram of Bubble Memory Controller

Key:

1. Control ROM and control signal former
2. Output buffer register
3. Control signals
4. Counter
5. Data line
6. Input/output buffer
7. Address line
8. Control lines
9. Control of functions
10. Registers
11. Data buffer
12. Page size control
13. Timing signals
14. Data input
15. Data register
16. Data output

The possibility of the loss of information when the power is cut off, when this information is in an interface register, is characteristic of the structure of a bubble memory chip with parallel-serial organization of the data array and a closed interface register. However, in the circuit of the controller presented in fig 2 the automatic return of information to storage registers and setting of the bubble memory to the initial ("zero") address upon a signal warning of the cutoff of power are provided for. Thus, comparison of floppy disk and bubble memory controllers demonstrates the considerable simplicity of a bubble memory controller, although at the same time great universality and the ability to perform more varied control functions are characteristic of it.

It is possible to organize in a bubble memory information and a sequence for accessing it by analogy with their implementation by means of a floppy disk memory. This, in fact, imitation makes it possible in the majority of instances to use already existing floppy disk memory controllers for controlling bubble memories. With this, of course, the ability to utilize such important advantages of a bubble memory as the parallel transfer of data, the start-stop mode, etc., is eliminated. On the other hand, a microcomputer user is able without special difficulty, at the first stage of using bubble memories (before the development or acquisition of a bubble memory controller) to successfully replace a floppy disk memory with a less expensive, faster, more reliable and more compact bubble memory because of the utilization of available controllers.

Slight modification of a floppy disk memory controller makes it possible to use it for controlling a bubble memory without imitating in the bubble memory the information structure of a floppy disk memory. With this approach the advantages of a bubble memory are utilized more completely. However, modification of the controller involves the introduction of changes in wiring and is impossible if the controller has been implemented as a large-scale integrated circuit. Standard microprocessors can act as controllers of an external bubble memory and for floppy disk memories. This application, as a rule, is justified when great universality and flexibility in controlling a memory are required.

Expansion of Microcomputer On-Line Memories

Microcomputers are usually used in the single-program mode, when for the purpose of increasing the effective capacity of the on-line memory in combination with high-speed semiconductor memories with random access (ZUPV's) it is possible to use less-high-speed memories. Let us discuss the feasibility of using as these devices bubble memories, which are distinguished by relatively low cost and relative simplicity of matching with a microprocessor. No less important is the fact that thereby the greater part of the microcomputer's on-line memory becomes independent of power.

The exchange of information between levels of such a 2-level on-line memory is accomplished through a single word. This considerably simplifies the structure of the bubble memory controller. As an example, in fig 3 is presented a structural diagram of a simple controller made by Rockwell International (USA) [6], which makes possible the exchange of information with a bubble memory in a single byte in the start-stop mode, as well as working with several bubble memory blocks. It is not difficult to see that the controller whose structural diagram is shown

in fig 3 is much simpler than the universal bubble memory controller illustrated in fig 2.

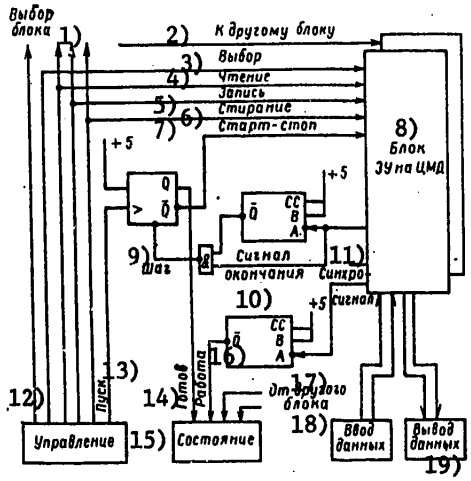


Figure 3. Structural Diagram of Bubble Memory Controller with Byte-by-Byte Exchange of Information: CC, B, A, > --flip-flop inputs; Q, $\overline{Q}$ -- flip-flop outputs

Key:

1. Accessing of block
2. To other block
3. Access
4. Readout
5. Write-in
6. Erase
7. Start-stop
8. Bubble memory block
9. Step
10. End signal
11. Synchronizing signal
12. Control
13. Start
14. Ready
15. State
16. Operate
17. From other block
18. Data input
19. Data output

The effectiveness of using a bubble memory for expanding an on-line memory can also be evaluated according to the criterion of minimizing the price of productivity, determined in this case as follows: $P_o = C_o T_o$, where C_o is the unit cost of the on-line memory and T_o is the access time.

When using a bubble memory together with a random-access memory, the expanded on-line memory represents a two-level hierarchy whose mean access time is determined by the expression in [7]: $T_o = T_z + (1 + h)T_{ts}$, where T_z and T_{ts} are the access times from the random-access memory and bubble memory, respectively, and h is the "percentage of successes," representing the relative share of accessing events satisfied by the first-level memory and depending basically on its capacity. With this the mean unit cost of the on-line memory can be computed by the equation: $C_o = (C_z N_z + C_{ts} N_{ts})/(N_z + N_{ts})$, where C_z, C_{ts}, N_z and N_{ts} are respectively

the unit costs and information capacities of the random-access memory and bubble memory. Thus, the price of the productivity of the expanded on-line memory will equal:

$$P_o = [(C_z N_z + C_{ts} N_{ts})/(N_z + N_{ts})]\,[T_z + (1 - h)T_{ts}] .$$

It is possible to find the optimum capacity of the random-access memory from the condition of ensuring a minimum for quantity P_o (maximum efficiency of the on-line memory), taking into account the dependence of index h on the information capacity, N_z, and the constancy of the capacity of the on-line memory, $N_z + N_{ts}$.

It is obvious that in many cases it can turn out that expansion of the on-line memory of a microcomputer by using a bubble memory will make it possible to increase considerably the memory's capacity with the same cost or to lower the cost with the same capacity while maintaining an acceptable overall speed of response.

Microcomputer Control Memory

The use of ROM's with the one-time write-in of information as control memories for microcomputers to a great extent determines their narrow specialization for the solution of a single specific problem. On the other hand, ever more often it has become effective to enlist microcomputers to solve not a single problem but some set of problems. In connection with this, devices which permit the repeated write-in of information, so-called reprogrammable devices [8], are needed as control memories. Random-access memories permitting the rapid write-in and readout of information and semipermanent memories with the erasure of information by means of ultraviolet light or by the electrical method [8] serve as reprogrammable memories suitable for operation as control memories for microcomputers. Bubble memories, distinguished by low unit cost, high information density and high reliability [1], can be used for the same purpose. As stated, the main disadvantage of a bubble memory in this application is the serial accessing of information and, as a result of this, the long random-access time. However, random access, characteristic of random-access memories, is not a typical requirement for control memories. Moreover, usually special software or hardware is used, e.g., an instruction address counter, which essentially makes possible the simulation of serial accessing, since the program is written into serial cells of the control memory. Thus, serial accessing can prove to be not so important a disadvantage of a bubble memory playing the role of a control memory.

The price of the productivity of the control memory of a microcomputer designed for running a certain set of programs can be determined as follows: $P_u = C_u T_u$, where C_u is the unit cost of the control memory, T_u is the access time for a single instruction, averaged by taking into account the program replacement time, and computed upon condition of the identity of the programs of the set by the equation: $T_u = T'_u + T_s/km$, where T'_u is the average instruction access time for a program, T_s is the time for replacing one program with another, k is the number of repetitions of each program and m is the number of instructions in a program.

Minimization of the price of productivity can serve as the criterion for selecting the type of control memory. The condition for greater efficiency of a control

memory constructed by using a bubble memory as compared with a random-access memory, for example, will be $P^{TsMD} < P^{ZUPV}$. Quantities T_u' and T_s when using a ZUPV [random-access memory] can be determined as follows: $T_u' = T_z$ and $T_s = T_z m$, and if a bubble memory is used as follows: $T_u' = (1 + B)/v_{ts}$ and $T_s = m/v_{ts}$, where B is a parameter characterizing the degree of branching of programs and equal to the product of the percentage of control transfer instructions and the mean difference between a control transfer instruction address and jump address (for the instruction to which the jump is made), and v_{ts} is the rate of transfer of data from the bubble memory.

Taking into account the expressions presented, the condition for the greater efficiency of a bubble memory as compared with a random-access memory is written as follows:

$$(C_{ts}/v_{ts})(1 + 1/k + B) < C_z T_z (1 + 1/k) .$$

From this inequality it is possible to determine the region of parameters characterizing the degree of repeatability, k , and branching, B , of programs of the set in which the use of a bubble memory as a control memory is more efficient than of a random-access memory. This region is determined by the equation:

$$B < [(C_z T_z - C_{ts}/v_{ts})/(C_{ts}/v_{ts})] [1 + 1/k] .$$

Reprogrammable memories of other types can be compared with bubble memories and regions of the parameters of the set of programs in which the use of bubble memories is more efficient can be found similarly.

It is of no minor importance that bubble memories can be used simultaneously as control memories and on-line memories for microcomputers [1, 9]. Furthermore, the use of a special instruction set can make it possible to interpret a bubble memory as a set of processor registers which results in considerable reduction of the size of programs and simplification of programming a microcomputer. In addition, the existence of this number of processor registers considerably simplifies the structure of the processor.

The studies presented in this article demonstrate that the state of the art of the development of bubble memories has occasioned the need to make a serious study of ways of more effectively using bubble memories in microcomputers. Furthermore, bubble memories designed on the basis of chips with serial and parallel-serial organization of the data array can find an application. It can be suggested that for designing microcomputer external memories large-capacity chips (a megabit and greater) with a parallel-serial structure are preferable. Furthermore, it can prove advisable to use low-capacity chips with a serial structure for expanding the on-line memory and especially for constructing the control memory.

Bubble memories constructed on the basis of existing bubble integrated microassemblies in the majority of cases cover in terms of capacity requirements for use in microcomputers. In this connection, the major direction for improvement of bubble memories oriented toward use in microcomputers will become, in our opinion, not increasing the capacity of chips, but increasing the speed of response of

devices. Furthermore, whereas at the present stage of development of bubble devices physical and technological limitations are to a great extent hampering progress in improving the clock rates of chips, in the future problems of finding new architectural solutions and information exchange algorithms conducive to improving the virtual speed of response of bubble memories can come to the forefront.

Bibliography

1. Naumov, B.N., Rayev, V.K. and Markarov, G.I. "Problems in Using a Bubble Memory in a Microcomputer" in "Magnitnyee elementy avtomatiki i vychislitel'noy tekhniki: XVI Vsesoyuznoye soveshchaniye; Tezisy dokladov" [Magnetic Elements of Automation and Computer Technology: 16th All-Union Conference, Theses of Papers], Moscow, Nauka, 1979.

2. Juliussen, J.E. COMPUTER DESIGN, Vol 15, No 10, 1976.

3. Fomina, I.M. "On the Question of Local Economic Criteria for Rating the Quality of Bubble Memories" in "Trudy INEUM, vyp. 69" [Works of the Institute of Control Computers, No 69], Moscow, 1978.

4. Peatman, J.B. "Microcomputer-Based Design," New York: McGraw-Hill, 1977.

5. Lee, D.M. "IEEE COMPCON, Spring 77, Digest of Papers," New York: IEEE Press, 1977.

6. Norton, R.R. "IEEE COMPCON, Spring 77, Digest of Papers," New York: IEEE Press, 1977.

7. Anacker, W. IEEE TRANSACTIONS ON MAGNETICS, Vol MAG-7, No 3, 1971.

8. Prangishvili, I.V. "Mikroprotsessory i mikro-EVM" [Microprocessors and Microcomputers], Moscow, Energiya, 1979.

9. Naumov, B.N., Rayev, V.K. and Markarov, G.I. "Bubble Memories for Microcomputers" in "Fizicheskiye svoystva i primeneniye tsilindricheskikh magnitnykh domenov v priborostroyenii" [Physical Properties and Application of Magnetic Bubbles in Instrument Making], Moscow, TsNIITEI priborostroyenii [Central Scientific Research Institute of Information and Technical and Economic Research on Instrument Making for Automation Equipment and Control Systems], 1979.

10. Bobeck, A.H., Blank, S.L., Butherus, A.D., Ciak, F.J. and Strauss, W. BELL SYSTEM TECHNICAL JOURNAL, Vol 58, No 6, 1979.

8831
CSO: 1863/43

UDC 681.327.66

FLAT MAGNETIC DOMAIN MEMORY MODULES WITH ADVANCED LEVEL OF INTEGRATION

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 10, Oct 81 pp 23-24

[Article by A.N. Ishchenko, engineer]

[Text] A trend in magnetic engineering involving the use of mobile magnetic domains as information media has received strong development in recent years here at home and abroad. The storage medium for devices of this class is a thin magnetic film with uniaxial magnetic anisotropy. Two different types of domain memories (DZU's) are distinguished depending on the orientation of the anisotropy vector: based on flat magnetic domains (PMD's) and cylindrical magnetic domains (TsMD's) [1, 6].

In the first kind of device the storage medium is a thin polycrystalline ferromagnetic film with an axis of easy magnetization (OLN) lying in the plane of the film, and in the second thin films and wafers of ferromagnetic single crystals with an OLN perpendicular to the plane of the film.

Characteristic to an equal degree of both kinds of devices are the properties of power independence and high reliability, caused by the absence of a moving mechanical medium; relatively low cost; and sufficiently high speed of response; which in many cases make their use extremely effective. Areas of application of DZU's include numerical program control equipment, equipment based on microprocessors, microperipherals, industrial robots, etc. The creation and use of DZU's will make it possible to fill the gap with regard to speed of response between a fast on-line memory and external storages for various kinds of computing systems, which in turn will increase to a great extent the execution rate of computer hardware.

However, in addition to these general features, flat magnetic domain memories have a number of important advantages over bubble memories: a simpler integrated technology, the absence of expensive raw materials, better speed of response owing to the ability to distribute several readout data transmitters over the memory plane and to organize parallel write-in and readout of information, the ability to function over an expanded temperature range, and cost which is more than an order of magnitude lower. All this makes it possible to recommend the extensive use of flat magnetic domain memories for instruments and automation equipment where the cost of equipment is low and the amount of output is considerable.

Operating Principle of Flat Magnetic Domain Devices

A thin polycrystalline film with induced uniaxial anisotropy of about 30 Oe and an OLN in the film's plane is used as the storage medium in which flat magnetic domains are maintained and move. These films are produced by vacuum deposition of a permalloy, in a constant magnetic field, onto a high-quality glass substrate consisting of borosilicate glass 0.2 to 0.3 mm thick.

For the purpose of restricting the movement of PMD's in a continuous film, special channels are formed which are surrounded by a mass having high coercivity. For the purpose of forming the structure of the storage medium it is possible to employ methods of diffusing a nonmagnetic material into magnetic films, chemical etching of the surface of films, the creation of irregularities on the surface of the substrate before deposition, deposition onto the substrate of a layer of nonmagnetic material, as well as a hard-magnetic film onto a soft-magnetic, etc.

Methods which have been developed at the present time for controlling the movement of flat magnetic domains in storage media can be divided into two classes: 1) methods based on the movement of unstabilized domains; and 2) methods based on the movement of domains stabilized by means of local magnetic fields.

Methods of the first class are characterized by the fact that the width of flat magnetic domains is predetermined by the width of the movement channels, and their length by the dimensions and spacing of control lines. The growth and interaction of domains are caused by the influence of pulses of the magnetic fields of the control lines. The unneeded part of a domain is erased by means of pulses of an erasing field, which nullify domains of reverse magnetization over the entire surface of the film, with the exception of just those sections protected by the effect of local confining fields which compensate the effect of erasure fields.

In devices of this class the preservation of flat magnetic domains is made possible in the absence of control fields only by the existence of coercivity on the part of the magnetic film storage medium. The minimum dimensions of stable flat magnetic domains are inversely proportional to the coercivity and for typical parameters of the medium (thickness of 1000 A, magnetization of 1000 Gs, anisotropy field of 30 Oe and coercivity of 3 to 4 Oe) equal a width of 5 to 7 μ and a length of 30 to 40 μ, which with the required stability margin corresponds to an information allocation density of about 10^4 bits/cm^2.

Methods of the second class are characterized by the fact that the retention and movement of flat magnetic domains are accomplished by a combination of variable control fields and stationary local stabilization fields. Studies [2, 3] have demonstrated the possibility of the existence of stable flat magnetic domains in external stabilizing fields with dimensions smaller than those occasioned only by the coercivity of the medium (for certain domain models and materials used the length of a domain is at least 10 μ_2 which corresponds to an information allocation density of about 0.25×10^6 bits/cm^2 and makes it possible to create memory modules with a capacity greater than 1 Mbit on a substrate measuring 30 X 48 mm).

The operating principle of devices of this class is explained by fig 1. Static stabilization fields induced by a special method along the channel (curve 1) and

the control line field (curve 2) are shown in fig 1a. At the starting moment, in the absence of a control field, the domain, under the influence of the stabilizing field, is at position I (fig 1b), and then under the influence of the total control and stabilization field grows to dimensions determined by the condition $H \geq H_t$ (position II), where H_t is the domain propagation field, which depends on the properties of the magnetic material of the medium and the geometrical parameters of the system. Then with reduction of the control field to zero the domain, under the influence of stabilization fields, is split into two domains (fig 1c), and under the influence of a control field of opposite polarity the right domain collapses and the left accomplishes further propagation through the structure of the channel (fig 1d).

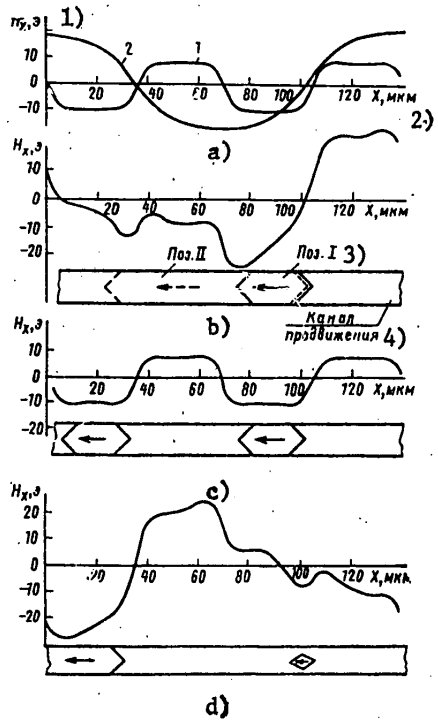


Figure 1.

Key:

1. H_x, Oe
2. μ^x
3. Position I
4. Propagation channel

Results of Investigation of Local Stabilization Fields

There are several practical methods which make it possible to form local stabilization fields: by means of current lines, by means of hard-magnetic overlays and by the creation of propagation channels with a special configuration [2, 5]. The first of these methods does not make possible power independence of a memory and is not described in this article. Fig 2 illustrates the idea of using stabilization fields formed by the second (a) and third (b) methods.

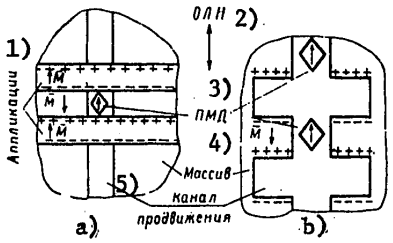


Figure 2.

Key:

1. Overlays
2. Axis of easy magnetization
3. Flat magnetic domains
4. Mass
5. Propagation channel

Let us discuss the second method. A periodic structure of hard-magnetic strips (overlays) magnetized opposite to the magnetization of the storage medium is placed at a certain distance from the storage plane. The strength of the stabilization field created by the overlays depends on the geometrical parameters of the system and the magnetic properties of the overlay material. The results of a theoretical study by the author of the magnetic fields of overlays for various distances, d , from the overlay plane to the storage medium are presented in fig 3. The parameters of the overlays are as follows: thickness--d = 500 Å, width--b = 35 μ, magnetization--M_s = 1432 Gs, coercivity--H_c = 150 Oe. In analyzing the distribution of the field it was assumed that the magnetization is distributed trapezoidally over the width of the overlay, which made it possible to reveal magnetic field inhomogeneities associated with the influence of the width of the edge zone.

Also, by means of computer methods, an analysis was made of the distribution of the magnetic field induced by a channel of special configuration without the use of hard-magnetic overlays. The results of calculation of the magnetic stabilization field for a magnetic film with the parameters M_s = 850 Gs , H_c = 40 Oe and d = = 1200 Å , where Y is the coordinate of the channel's width, are presented in fig 4.

A comparison of these two methods of forming stabilizing fields makes it possible to rate their advantages and disadvantages. The advantage of the second method is the simpler technological process whereby the channel's configuration is formed

simultaneously with the creation of a high-coercivity mass, and its disadvantage as compared with the first method is the greater inhomogeneity of the stabilization field over the width of the channel, which necessitates additional consideration of the properties of flat magnetic domains in inhomogeneous fields.

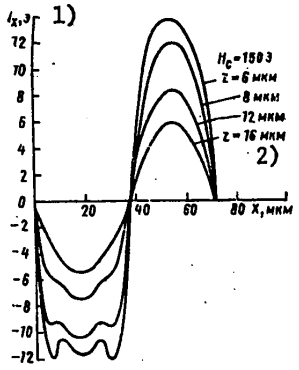


Figure 3.

Key:

1. H_x, Oe
2. μ

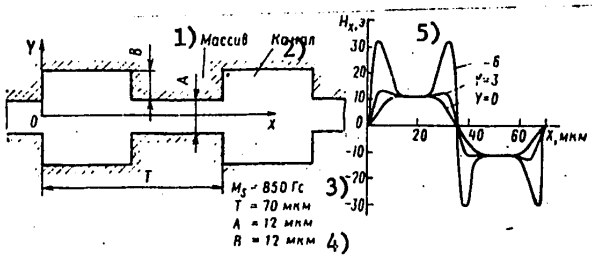


Figure 4.

Key:

1. Mass
2. Channel
3. 850 Gs
4. μ
5. Oe

Results of Investigation of Elements of Closed Registers

The designs of flat magnetic domain memory matrixes with local stabilization fields which have been implemented at the present time are characterized by the presence of unclosed registers representing a system of straight-line low-coercivity channels in which the number of readout data transmitters equals the number of registers and the copying of information is performed by means of electronic regeneration circuits [2]. However, in a number of applications it is more effective to use long closed

registers, which makes it possible to reduce considerably the number of terminals in a matrix, to simplify the circuit and to lower the cost of the memory control unit.

In this connection in designing flat magnetic domain memory matrixes of great importance is the consideration of structural elements implementing turning of a low-coercivity channel and of the magnetic fields associated with the geometry of these elements. Taking into account the leakage fields created by the edges of the high-coercivity mass makes it possible to simulate properly the passing of a flat magnetic domain through a structural "turn" element and to determine the parameters of the structure required for obtaining a sufficiently broad range of stable operation. An analysis was made of a number of element configurations in the research process. As a result it was established that the configuration illustrated in fig 5, where the distribution of the magnetic leakage field in this element is given, is characterized by the greatest inhomogeneity of leakage fields. The parameters of the magnetic film medium correspond to the example of calculating stabilization fields in a channel of special configuration. The field in the "turn" is a brake field and therefore at the moment the flat magnetic domain passes through the turn an additional compensating field is required, created by the control line in the direction of the axis of easy magnetization. A fragment of the topology including the key elements of a shift register--the shifting element and the turn with control lines--is illustrated in fig 6.

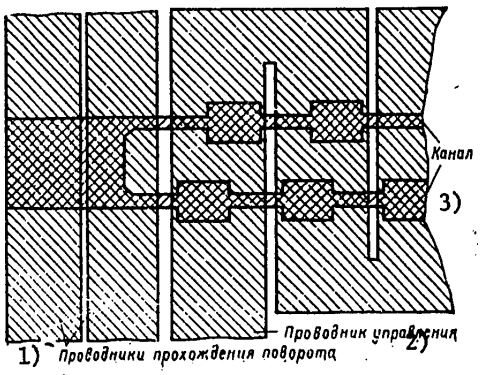


Figure 5.

Key:

1. Conductors for passing through turn
2. Control conductor
3. Channel

Conclusion

The theoretical and experimental studies carried out make it possible to determine the required geometrical dimensions and the relationships between the basic elements of flat magnetic domain memory matrixes and local stabilizing fields for

specific magnetic parameters of the storage medium, as well as, by taking into account the power relationships obtained in [3], to optimize them and create matrixes with an advanced level of integration.

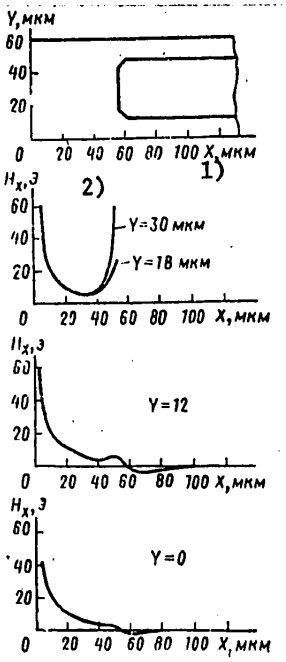


Figure 6.

Key:

1. μ
2. Oe

Bibliography

1. Boyarchenkov, M.A., Vasil'yeva, N.P. and Rozental', Yu.D. "Logicheskiye ustroystva na magnitnykh sredakh s upravlyayemym dvizheniyem domenov" [Logic Devices Employing Magnetic Media with Controlled Motion of Domains], Moscow, Energiya, 1978.

2. Battarel. C.P., Marille, R. and Nori, J.P. IEEE TRANS. MAGN., Vol MAG-13, 1977, pp 1302-1304.

3. Vasil'yeva, N.P. and Semenov, V.S. "Stability of Flat Magnetic Domains," AVTOMATIKA I TELEMEKHANIKA, No 8, 1980.

4. Sukhu, R. "Magnitnyye tonkiye plenki" [Thin Magnetic Films], Moscow, Mir, 1967.

5. Patent No 2406284 (France).

6. "Prospects for the Application of Magnetic Films with Controlled Motion of Domains in Computer Technology Devices, a Thematic Selection," PRIBORY I SISTEMY UPRAVLENIYA, No 6, 1973.

8831
CSO: 1863/43

UDC 681.3:531.78.2

INTERFACING OF DIGITAL STRAIN GAGE BRIDGES WITH 'ELEKTRONIKA-60' MICROCOMPUTER

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 10, Oct 81 p 32

[Article by V.D. Uspenskiy and M.G. Engovatov, engineers, and A.E. Kharlap, candidate of physical and mathematical sciences]

[Text] For repeated measurements of static strains and stresses originating in mechanical elements and structures extensive use is made of digital strain gage bridges of the TsTM-3 and TsTM-5 type with recording of the results on a printer or punch. The further processing of measurement results is performed on keyboard or general-purpose computers.

At Giprouglemash [State Planning and Design and Experimental Institute of Coal Machine Building] (Moscow) a unit has been developed for interfacing digital strain gage bridges with an "Elektronika-60" microcomputer for the purpose of combining the processes of recording and processing experimental data.

The interface (US) (fig 1) includes matching switches (SK's) required for transforming the levels of the signals of digital strain gage bridges (TsTM's) to standard form; level converters (PrU's) which convert negative logic into TTL logic; a code converter (PrK) which converts binary-decimal code into binary according to an algorithm for dividing the binary-decimal number by two with carryover of the lower-order bit to the corresponding bit of the binary number; and a control circuit (SkhUpr) which starts TsTM bridges, controls the PrK converter and swaps service signals with the computer. The interface is linked with the "Elektronika-60" computer by means of an I1 parallel exchange board.

When the computer is turned on and the program is started the processor generates a "Reset B" signal which sets flip-flops T1 and T2 in the control circuit to the initial state (fig 2).

Before each measurement cycle the computer forms an "Output B" instruction which triggers former F5. The length of the pulse in the latter's output is determined by the time for the operation of electromagnetic relay R, whose contacts control the channel switch (PK) (cf. fig 1) and start the TsTM bridge. At the end of the measurement time the TsTM bridge issues an "End Conversion" signal (KPr) to trigger former F1, by means of its pulse flip-flop T1 is set to the "1" state and the pause former, F2, is triggered. With this the appearance of a signal for enabling

the operation of the code converter (RPr) from the "1" arm of flip-flop T1 enables the operation of converter PrK (cf. fig 1). The duration of the "Pause" pulse is determined basically by the time lag of the relay unit of the TsTM bridge, as well as by the time delay for the operation of converter PrK and equals 50 to 60 ms.

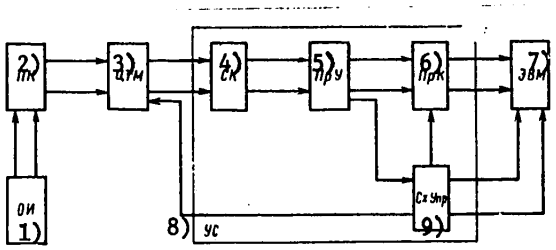


Figure 1. Block Diagram of Automated Strain Gage System: OI--test object

Key:

1. Test object
2. Channel switch
3. Digital strain gage bridges
4. Matching switches
5. Level converters
6. Code converter
7. Computer
8. Interface
9. Control circuits

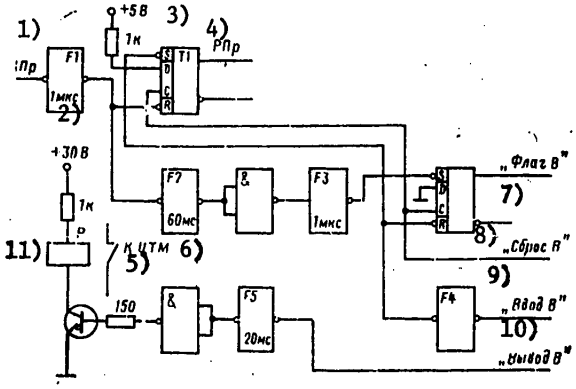


Figure 2. Circuit Diagram for Control of Digital Strain Gage Bridges and Input of Information Into the Computer

Key:

1. "End Conversion" signal
2. μs
3. + 5 V
4. Code converter operation enabling signal
5. To digital strain gage bridge
6. ms
7. "Flag B"
8. "Reset B"
9. "Input B"
10. "Output B"
11. Relay

At the trailing edge of the "Pause" pulse former F3 is triggered, setting flip-flop T2 to the initial state, informing the computer that the measurement cycle has been completed.

With the appearance of the "Flag B" signal the computer forms the "Input B" instruction, with which readout of the measurement result takes place and former F4 is triggered and flip-flops T1 and T2 are reset by the latter to the initial state, after which the measurement cycle is repeated. The interface control circuit has been executed with series K176 microcircuits. The data accumulated are processed by means of an appropriate program with the output of processing results to a computer printer.

The interface is designed as a completed structure on whose frame are installed connectors of the RPPM182288GYe0.364.209 TU type. Connections are installed by the twist-on method.

No changes have been introduced into the circuit diagram of the digital strain gage bridge, which makes it possible to use the instrument independently.

8831
CSO: 1863/43

UDC (53.085.3:65-52).004.1

SIGNAL PANEL FOR AUTOMATED CONTROL SYSTEM

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 10, Oct 81 pp 32-33

[Article by A.M. Frid, engineer]

[Excerpts] In an ASU [automated control system] it is necessary to present personnel with on-line information regarding the state of the controlled system. In the case of binary information it is a good idea to use a 2-position signal panel for its representation.

At TsNIITU [Central Scientific Research and Design and Technology Institute of Control Organization and Equipment] (Minsk) a signal panel has been developed for the Minsk Tractor Plant's production ASU.* It has been installed at the plant controller's station and serves the purpose of representing the state of the process stock of completion items along the intershop part manufacturing production process route and on a shop's assembly line. The panel is connected to the information source (a YeS-1030 computer) by means of an RI-8901 remote data acquisition unit. With the failure of this communication channel the input of messages to the panel is performed from an RI-7501 data recorder situated at the assembly shop controller's station.

The panel can be connected to series YeS and SM computers via modifications of equipment of the APD-21S type, designed for operating with these computing complexes. For connection of the unit to APD [data transmission equipment] of another type or directly to a computer it is sufficient to replace in it the APD interface.

The panel's field consists of unified removable mosaic elements measuring 40 X 40 mm produced by the Zhitomir Promavtomatika Plant.

Technical Data of Panel

Number of signal cells	127
Maximum distance between panel and information source, km	10
Rate of reception of data in characters per second	100
Distance for reliable reading of display, m	8
Size of field, mm	1200 X 1200

*Velesko, Ye.I. and Dudkin, G.Ye., "Automation of Intershop On-Line Real-Time Control," PRIBORY I SISTEMY UPRAVLENIYA, No 6, June 1979.

8831
CSO: 1863/43

UDC 681.14:681.616.065

ALPHANUMERIC COMBINED KEYBOARD FOR READOUT OF CHARACTERS IN 16-BIT POSITION CODE

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 10, Oct 81 pp 33-34

[Article by V.N. Borovskikh, D.N. Pospelov, L.V. Pikalevskiy, A.I. Martynenko and N.Kh. Sivets, engineers]

[Text] At NII UVM [Scientific Research Institute of Control Computers] (Severodonetsk) a set of keyboards has been developed for inputing information into a computing complex, designed for use as part of the work site of an industrial engineer operator. They differ in character set, structure and purpose. The principle of coding characters is common to all these keyboards. A description of one keyboard of the set is given below.

The key area of the keyboard, containing 128 keys, is divided into groups of 16 keys each (groups can be incomplete) according to the principle illustrated in fig 1. The keyboard is designed with keys without holding. A given circuit accesses one group of keys, which makes it possible to determine the state of keys of only the polled group (OG). From the processor a word arrives containing the number of the OG in binary code, which is stored in a buffer register and is decoded by means of a 4-bit decoder. A low-level signal appears in the input line of the OG and if thereupon a key has been pressed a low-level signal is formed in the corresponding bit of the word arriving in the processor. Thus, information obtained from the processor regarding accessing of the group and information transferred to the processor regarding the bit relating to the key pressed make it possible for the microprogram to code the transferred character by means of an appropriate code, e.g., KOI-7, KOI-8 and DKOI [EBCDI] code, using code expansion facilities in keeping with CEMA Standard ST SEV 360-76.

In the exchange of information a zero signal must be in only one bit of the transferred word. The appearance of two or more signals means two or more keys have been pressed (i.e., an operator error), is recognized by the processor as an error and accordingly is not coded.

This organization of the keyboard makes it possible to increase the number of polled groups to 16 and accordingly the number of keys to 256, and when necessary with an increase in the word length of the buffer register and decoder, to any required number. It is possible to connect several keyboards to a single control circuit.

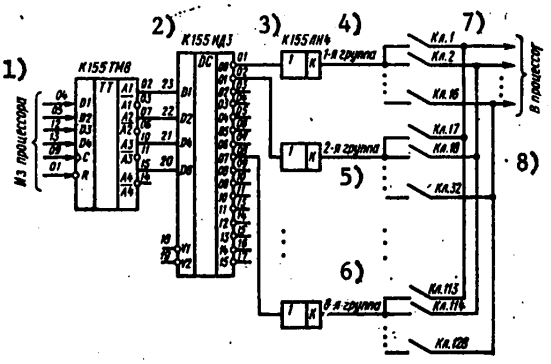


Figure 1.

Key:

1. From processor
2. K155ID3
3. K155PN4
4. First group
5. Second group
6. Eighth group
7. Keys
8. To processor

Because of the absence of keys with mechanical holding, for mode and control keys a light signaling system has been added. The switching of this system on and off is program controlled and the pressing or release state is stored in the keyboard's flip-flop memory. The light signaling system control circuit is presented in fig 2. When information is obtained on the pressing or release of mode or control keys, in the processor a binary code is formed for turning on the appropriate lamp element; the number of lamp elements also can be increased by increasing the control elements.

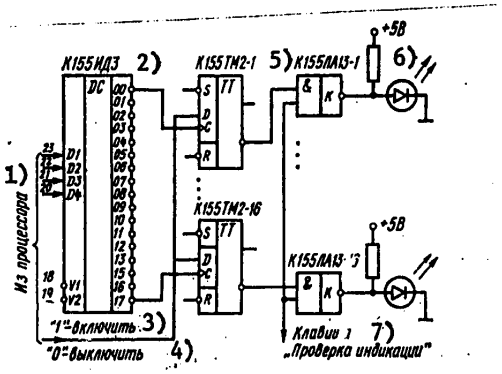


Figure 2.

Key:

1. From processor
2. K155ID3
3. "1"--on
4. "0"--off
5. K155PA13-1
6. + 5 V
7. "Check Display" key

For the purpose of indicating to the operator working with the keyboard specific actions, an electrodynamic loudspeaker is provided in it, which is program controlled (cf. circuit in fig 3). Interfacing and the exchange of information between the keyboard and processor are performed by means of a standard IUS interface (object interface unit).

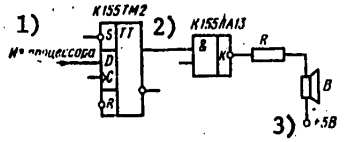


Figure 3.

Key:
1. From processor
2. K155PA13
3. + 5 V

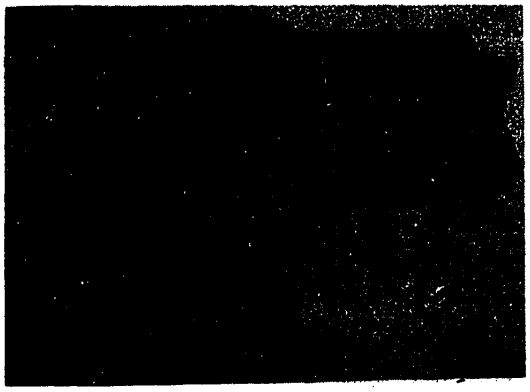


Figure 4.

Structurally the keyboard consists of a keyboard unit in the form of a desktop or built-in unit containing a key area, lamp elements and an electrodynamic loudspeaker, and of a connecting cable assembly and control unit connected to the processor. The general appearance of the keyboard is shown in fig 4. The small number of equipment units, utilizing microcircuits, and the simplicity of the keyboard unit have made possible high performance characteristics and reliability for the keyboard.

8831
CSO: 1863/43

UDC 621.316.541

MINIATURE VIBRATIONPROOF MULTIPOINT ELECTRICAL CONNECTOR*

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 10, Oct 81 p 36

[Article by G.V. Smirnov, engineer]

[Text] Because of the extensive use of printed circuitry in electrical measuring equipment there has been a drastic increase in the need for connectors which interconnect functional units executed on a single printed circuit board.

At the special design bureau for testing and control equipment at the Krasnoyarskiy ZIP [Measuring Instrument Plant] Plant a design, which is simple in terms of labor intensiveness (all parts are fabricated on automatic machines), of an intraunit connector has been developed and is being used successfully in high-precision electrical measuring instruments.

A connector in the position of the connection of the current-carrying tracks of a printed circuit board with it (a) is shown in fig 1 and the positions of the disconnection (b) and connection (c) of the printed circuit board with the connector are shown in cross section.

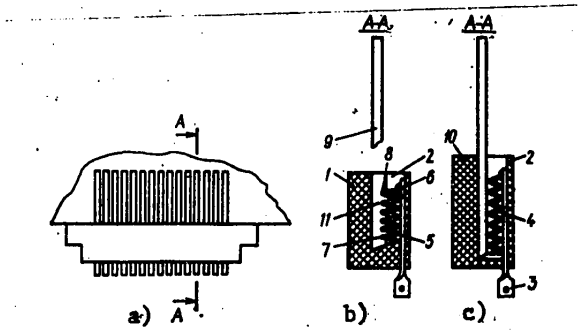


Figure 1.

*USSR Patent No 538440.

The electrical connector, 10, includes a case, 1, made of a molded insulating material of the ABS-220 type. In it on the side of the connection of the reciprocal half there is a common slot from which cells, 2, run to one side over the entire depth. In each of them is placed a stiff contact, 3, made in the form of a current-conducting flat piece fastened in a slot in the base of the case, 1. The contact, 3, adjoins on one side the side wall of the cell, 2, and on the other the outside diameter of a coiled spring, 4. Ends 5 and 6 of the coiled spring, of end turns 7 and 8, are fastened to contact 3, for which in the contact there are notches made at the same time the contact was punched out, and the axis of the spring is parallel to the plane of the contact.

The reciprocal half, 9, of the connector is an element of the board's general circuitry. It is made of a thin sheet of glass-cloth-base laminate or paper-base laminate with current-conducting tracks 2 mm wide with a spacing of 3.75 mm.

Prior to connection of the reciprocal half to the connector, each of the turns of the coiled spring is in the free state and is situated perpendicular to the plane of the contact and makes reliable contact with it with its outside side surface. Between the side wall of the slot opposite the contact and the coiled spring a gap, 11, is formed, into which the reciprocal half of the connector is inserted, and its width is always smaller than the thickness of the reciprocal half. When the latter is connected to the connector the distance between the current-conducting tracks of the reciprocal half and the contact plane of the contact becomes less than the outside diameter of the spring. In this case each turn of the spring is forced to occupy an oblique position relative to its initial position and because of its elastic properties and tendency to return to its initial position ensures a reliable electrical contact by means of its outside surfaces.

The number of turns can be made rather great (n = 15 to 20 without any increase in overall dimensions) and thus it is possible to form a highly reliable electrical contact with minimum variation in contact resistance. For example, on the basis of a coiled spring with an outside diameter of 2.8 mm, a bronze wire diameter of 0.2 mm, a number of turns equal to 10, thickness of the flat contact equal to 0.3 mm, thickness of the reciprocal half equal to 1.5 mm and a working gap 0.5 mm wide, static variation of contact resistance equal to 0.5 MΩ and dynamic of 1 MΩ are made possible, as well as high reliability of contact over a long time, including under conditions of jolting and vibration. The introduction of this design has made it possible to gain a savings of 1.444677 million rubles at the Krasnodarskiy ZIP Production Association and to improve considerably the reliability of instruments.

8831
CSO: 1863/43

UDC 658.562.6:002.56

AUTOMATION OF TESTING AND ADJUSTMENT OF HYBRID INTEGRATED CIRCUIT RESISTORS WITH COMPUTER CONTROL

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 10, Oct 81 pp 39-41

[Article by S.V. Karavaykin, V.S. Punkevich, O.N. Selyutin and V.P. Umnov, engineers]

[Text] The most important and labor-intensive operations in the production of hybrid integrated circuits are the testing and rejection of microcircuits on a substrate and the adjustment of passive elements in each microcircuit. At the present time the trend has been observed of developing combined technological equipment making it possible to combine these operations.

At the Penza branch of VNIITIpribor [All-Union Scientific Research Technological Institute of Instrument Making] an automated unit of the AMTs 0637 type has been created for testing and adjusting resistors with control by a programmable keyboard computer (PEKVM) of the "Iskra-125" type. In the unit is used the method of adjustment based on removal of a part of the resistive layer by means of a laser beam, because of which the resistor's resistance is increased.

The accuracy and efficiency of adjustment depend not only on the equipment used, but also on the configuration of the section removed [1]. Therefore, software is provided in the unit for making cuts of three types: L, Y and U (fig 1a to 1c, respectively). An L-cut is used in cases when it is necessary to have high efficiency and an energetically optimum configuration of the resistor after adjustment with relatively low precision (0.5 percent); a Y-cut makes possible high adjustment precision (not worse than 0.05 percent) and the ability to adjust resistors whose layer is intersected by a contacting probe; a U-cut is used when the topological dimensions of the microcircuit deviate from the nominal by more than 0.05 mm.

The amount of cross cutting, t_i, is computed from the equation

$$t_i = b_i \Big/ \left(1 - \eta + \frac{\eta}{\delta R_x}\right) + C, \qquad (1)$$

where b_i is the width of the resistor to be adjusted or of the additional adjustment section; C is the program amount of correction, taking into account

the zone of recession from the contact areas of the resistor and the configuration of the cut (the value of quantity C is negative for L- and U-cuts and positive for a Y-cut); δR_x is the relative deviation of the resistor's resistance from the nominal value of R_{nom} ; and $\eta = R_{dob\ [additional]}/R_{nom}$; (R_{nom} is the calculated value of the resistance of the part of the resistor to be adjusted).

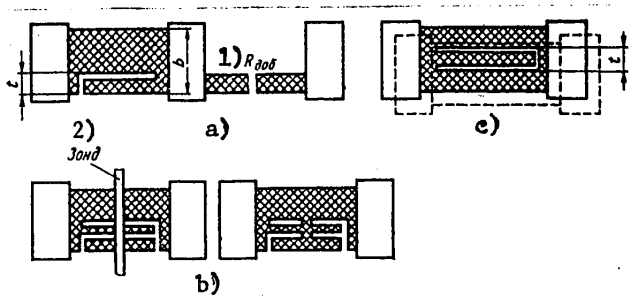


Figure 1. Configuration of L- (a), Y- (b) and U-Cuts (c)

Key:
1. R_{dob}
2. Probe

In adjusting non-compound rectangular resistors equation (1) is simplified, $\eta = 1$, and assumes the form:

$$t_i = b_i \delta R_x + C \ . \qquad (2)$$

A structural diagram of the unit is presented in fig 2.

The "Iskra-125" computer is intended to control the operation of the system and the measuring and actuating units included in it, for enabling the exchange of information between functional blocks and units of the system, and for processing information arriving from functional blocks and units.

In order to increase the capacity of the on-line memory, equal to 1K bytes, the "Iskra-125" PEKVM has been given an additional cassette memory with a capacity of 80K bytes.

The interface unit serves the purpose of 2-way conversion of information circulating between the "Iskra-125" PEKVM and functional blocks and units of the apparatus, which has special-purpose sets of information and control lines, as well as for executing certain logic and control operations directed toward increasing the unit's speed of response.

The most important part of the unit determining its metrological characteristics is the type AMTs 1442 analyzer. The analyzer consists of a block of standard resistors (BOR) which upon instructions from external units forms nominal values of the

resistance of the resistors to be adjusted in the range of 10^2 to 10^6 Ω, and of a testing (BK) and processing (BO) unit.

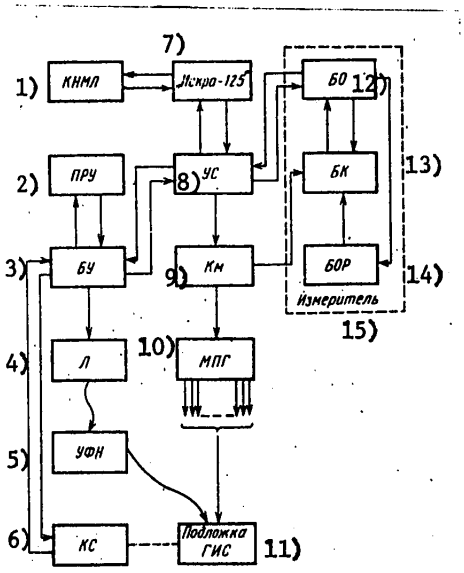


Figure 2. Structural Diagram of Unit: KNML--cassette memory; BU--unit for controlling motors of coordinate table and laser; L--laser; UFN--focusing and observing unit; US--interface

Key:

1. KNML
2. Manual control panel
3. BU
4. L
5. UFN
6. Coordinate table
7. "Iskra-125"
8. US
9. Switch
10. Multiprobe air-floated head
11. Hybrid integrated circuit substrate
12. Processing unit (BO)
13. Testing unit (BK)
14. Standard resistor block (BOR)
15. Analyzer

The BK unit includes tw key elements: a comparator (K) and a resistance-deviation-to-time-interval converter (POSV).

The comparator is intended to test the resistance of a resistor in the adjustment mode. Its measuring circuit is in the form of a balanced bridge two arms of which are formed by a code-controlled divider which specifies the percentage deviation from the nominal value of the resistor to be adjusted and the other two are standard resistors and the resistor to be adjusted. The moment of equilibrium of the bridge ensues with equality of the value of the resistor being adjusted to the nominal value with the specified percentage deviation. This moment is recorded by means of a comparison unit whose output signal is averaged for the purpose of eliminating the

influence of impulse noise. The total comparison time is not longer than 100 μs. The comparison error is not greater than ± 2 percent.

The resistance-deviation-to-time-interval converter makes possible operation of the analyzer in the precise adjustment and measurement mode and is in the form of an integrating percentage-deviation-to-time-interval converter based on the unit described in [2].

The total measurement time in the measuring and precise adjustment modes is not greater than 55 ms. The relative conversion error in the 10^3 to 10^5 Ω range is not greater than ± 0.05 percent and in the 10^2 to 10^3 and 10^5 to 10^6 Ω ranges not greater than ± 0.1 percent. The power dissipated in the resistor adjusted is not greater than $5 \cdot 10^{-3}$ W.

The BO unit represents a unit which forms pulse trains with the frequencies required for the functioning of the comparator and POSV circuits, as well as for processing the POSV's output signal and indicating the measurement result in digital form.

The analyzer's inputs are connected to the resistor being tested via a switch (KM) and a multiprobe air-floated head (MPG). The latter is not rigidly fastened to the coordinate table; therefore, adjustment can be performed on substrates with any number of microcircuits. Because of this characteristic of the head it is possible rapidly to readjust the unit for a microcircuit of another type by changing boards with probes.

The coordinate table (KS) is moved along axes X, Y and Z by means of motors controlled by unit BU. A circuit of amplifiers with electronic boosting of the phase current of the stepper motors is used in this unit, which made it possible to lower the electric power required by the apparatus.

The apparatus operates in the following manner. The operator places on the stage the hybrid integrated circuit substrate which is to be tested and by means of a manual control console (PRU) places the first microcircuit at the initial adjustment point, checking the correctness of this placement by means of the cross-hairs of the optical system. With this the preparatory operations are concluded and the operator by pressing the SPr (Program Calculation) key on the "Iskra-125" PEKVM's console switches the apparatus to the automatic operating mode.

The unit's software makes it possible to program the adjustment starting points and to select automatically the optimum by-path for hybrid integrated circuit resistors when adjusting according to the results of tolerance testing.

Technical Data of Apparatus

Error in adjustment of resistors, percentage	± (0.1 to 0.05)
Resistance range of resistors which can be adjusted, Ω	$1 \cdot 10^2$ to $1 \cdot 10^6$
Maximum number of resistors which can be adjusted on a single microcircuit, units, not greater than	30
Comparison time in ms, not greater than:	
Rough	0.2
Precise	60

Maximum size of area of board on which it is possible to adjust resistors, mm, not greater than	60 X 48
Intervals in which substrate can be moved, μ.	5 ± 2
Rate of movement of substrate, mm/s	0.04 to 8
Travel of stage along axes X and Y, mm, not greater than	20
Number of fixed postions of stage along axis Z	3
Monochromatic radiation source	LGI-21 laser
Width of cut, μ, not greater than	30
Program medium	MK-60 cassette with magnetic tape
Continuous operation time, h, not more than	8
Supply voltage, V	380 ± 10 percent
Power requirement, kW, not greater than	1.5
Overall dimensions, mm	2500 X 750 X 1270
Weight, kg	300

Bibliography

1. Selyutin, O.N. and Umnov, V.P. "Analysis of Methodical Errors in Adjustment of Film Resistors of Microcircuits" in "Tekhnologicheskiy kontrol' v priborostroyenii" [Technological Testing in Instrument Making], Moscow, VNITIpribor, 1977.

2. Frolov, V.M. and Andreyev, A.B. "Integrating Voltage-to-Time-Interval Converter," PRIBORY I TEKHNIKA EKSPERIMENTA, No 4, 1979.

8831
CSO: 1863/43

UDC 681.327

ACCESSORY MODULE FOR INPUT OF INITIATIVE SIGNALS

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 pp 8-9

[Article by I.I. Popov and G.S. Pyatlina, engineers]

[Excerpts] The digital data input modules (MVvDI's) and initiative signal input modules (MVvIS's) widely used in the structure of M-6000 and M-7000 process control computers do not always meet the demands of users. The absence in the inputs of these modules of memory elements and of blocking of the entry of new data while modules are being polled limits their application in the input of pulse data. Therefore, with a high frequency of the entry of input signals, when each pulse carries information, a loss of information takes place while data transmitters are being polled and this restricts the range of application of these modules.

An initiative signal input module which is without these disadvantages has been developed at the Karelian Scientific Research Institute of the Foresty Industry (Petrozavodsk) for the purpose of the acquisition of data from pulsed data transmitters entering a process control computer for processing. The MVvIS proposed is designed to receive, normalize, store and input into a process control computer a parallel 8-bit binary code. The module can be polled both by program and upon the module's initiative.

Main Specifications of Module

Number of input channels	8
Level of input signals	Corresponds to levels of signals of A622-4 standard module
Possibility of input from contact pickups	Yes
Number of modifications of module depending on type of input signals	14
Maximum frequency of input signals in kHz	200
Capacity of memory for each input in bits	1
Formation of interrupt signal	When any register location is filled
Interface rank	2K
Supply voltage in volts	5 ± 0.25
Current requirement in mA, not more than	400

The use of an accessory MVv1S makes it possible to improve the specifications and to expand the functional capabilities of process control computers designed on the basis of M-6000 and M-7000 processors.

8831
CSO: 1863/80

UDC 681.337

SYSTEM FOR INPUT OF ANALOG SIGNALS INTO M-6000 COMPUTER

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 pp 9-10

[Article by A.N. Konotopov and V.S. Nikul'shin, engineers]

[Excerpts] In the creation on the basis of the hardware of the M-6000 ASVT [modular computer hardware system] of ASUTP's [automated systems for controlling technological processes] for systems with a high level of electrical noise, e.g., electric arc units, special difficulties arise in measuring low-level signals transmitted over distances greater than 15 m.

The output signal of sensors such as thermocouples (types KhK, KhA and VR5/20), resistance thermometers, etc., under working conditions is not greater than 25 to 35 mV. Experience has demonstrated that the electrical interference of a noise nature in lines for the electrical connection of primary transducers with the input of a computer can reach 0.5 V. The ratio of the amplitude of noise to the range of the input signal is 15 to 200, which exceeds the permissible value of 10 for the A613-6 filter module [1]. In addition, high operating currents of automated systems are dangerous with regard to the appearance of an additional difference in potential (10 to 20 V) between the grounds of units and the ground of the computer. This difference in potential causes additional longitudinal noise, which in a number of cases can become the reason for computer units' failing, since there is no galvanic input-output isolation in the M-6000 computer.

An attempt to use the commonly known means for suppressing noise and interference in such an unfavorable case has not resulted in success [2]. Such a method of solving the problem as installing in the direct vicinity of each primary transducer an amplifier-normalizer with galvanic input-output isolation, e.g., of the NP-5-B1 type, with a large amount of data to be picked up becomes quite expensive and requires additional space.

For purposes of reducing the expense of creating an ASUTP we have proposed that a single amplifier be used for amplifying a group of identical signals. In this case the layout of the system for the input of analog signals (the SVAS) into an M-6000 computer has been changed. Its structural diagram is presented in fig 1.

Normalizing converters of the NP-5-B1 type with appropriate calibration are used as amplifiers in the system, which have galvanic input-output isolation and multistage filter discrimination. The converters selected can each operate with

a group of sensors of the same type numbering up to 64. The time for NP-5-B1 converters to reach their rating from the moment of the end of the switching of input circuits is not greater than 80 ms [3]; therefore, the real switching time for a single point is close to 100 to 120 ms. The BVVr-1 time delay unit was reset in order to make possible this delay in the switch control module [4].

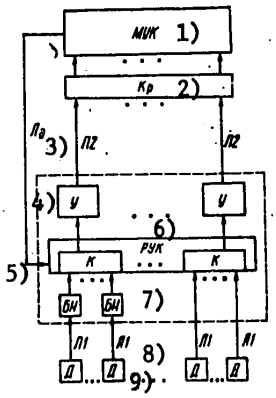


Figure 1.

Key:

1. Switch control module
2. Cross bay
3. Additional line
4. Amplifier
5. Switch
6. Switch control expander
7. Normalizing unit
8. Line 1
9. Sensor

It is also feasible to input medium-level signals via switches of the first switching stage, taken out to the automation system.

The SVAS described makes it possible to reduce the number of amplifiers required (10- to 16-fold) and the number of lines for electrical connection of the automation system to an M-6000 computer.

An ASUTP which has been developed and introduced for an electric arc unit with the SVAS described in its structure makes it possible to gather and process information from 17 frequency signal pickups and 146 analog signal converters, whereby 84 of these converters have low-level signals in their output. A total of eight amplifiers is used in the low-level SVAS. Since only medium-level signals are supplied to the inputs of the second switching stage, it was possible to implement group polling with a polling time for all 146 primary analog signal converters of not greater than 2 s.

The use of this SVAS as part of the structure of an ASUTP for an electric arc unit over the course of a year demonstrated its high reliability and sufficient noise immunity. The noise immunity level of the system described is witnessed by the fact that the noise level in the input of the second switching stage of the M-6000 computer is not greater than 50 mV.

The layout proposed for the SVAS is especially advantageous in the creation of ASUTP's for systems characterized by a great number of controlled parameters and which are at distances of greater than 15 m from the computer.

Bibliography

1. "Komponovka sistemy vvoda analogovykh signalov M-6000: Instruktsiya" [Layout of System for Input of Analog Signals for the M-6000: Instructions].

2. Ott, G. "Metody podavleniya shumov i pomekh v elektronnykh sistemakh" [Methods of Suppressing Noise and Interference in Electronic Systems], translated from English, edited by M.V. Gal'perin, Moscow, Mir, 1979.

3. "Preobrazovateli izmeritel'nyye bystrodeystvuyushchiye NP-5-B1, NP-5-B2, NP-5-B3: Tekhnicheskoye opisaniye i instruktsiya po ekspluatatsii OYaa.140.504TO, 1975" [NP-5-B1, NP-5-B2 and NP-5-B3 High-Speed Measuring Converters; Technical Description and Instructions on Use, OYaa.140.504TO, 1975].

4. "Modul' upravleniya kommutatorami A612-1: Rukovodstvo po ekspluatatsii 2.320.002RE,2.320.002RE2" [A612-1 Switch Control Module; Use Manual 2.320.002RE,2.320.002RE2].

8831
CSO: 1863/80

UDC 53.087.62:681.32

Shch68400 AND Shch68000K PRINTERS

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 pp 27-28

[Article by O.V. Kasparova and T.G. Provorova, engineers]

[Text] The Shch68400 unit is designed for the automatic recording on paper tape of information entering from digital measuring instruments and other information sources.

Specifications of Shch68400 Unit

Printing speed in lines per second	40
Number of:	
Digits	16
Printed characters	16*
Spacing in mm:	
Between digits	2.54 ± 0.5
Between lines	4.23 ± 0.5
Field of printed characters in mm:	
Height	2 to 2.5
Width	1.3 to 1.6
Discrepancy in height of characters in a line, in mm, not greater than	± 0.5
Overall dimensions in mm	175 X 380 X X 490
Weight in kg	20
Supply voltage in V	220 +10, -15 percent (50 Hz ± 1 percent)
Power requirement in V·A	190
Tentative price in rubles	3000

*0 to 9, = , - , + , space, > and < .

The unit contains an interchangeable interface, has semiautomatic setup of paper tape and makes possible the display and printout of recording time.

The external appearance of the instrument is shown in fig 1.

Figure 1.

The Shch68000K unit (fig 2) is designed for the automatic recording on paper tape of information arriving in 1-2-4-8 parallel binary-coded decimal code from digital measuring instruments and other information sources.

Figure 2.

The unit is produced in keeping with TU [Specifications] 25-04-3018-75 and contains a printing mechanism and a transcriber.

The printing mechanism consists of a character wheel unit, code electromagnets, a paper transport, an inking unit, an electric motor and an induction-type synchronizing generator.

The transcriber includes stabilized power supplies, printing electromagnet drive and paper transport amplifiers, an electric motor operating mode control unit, a buffer storage for a single line of printed information and matching units.

Printing is accomplished by means of the impact of printing strikers on the paper with the character drum constantly rotating.

The operating principle of the unit consists in synchronized control of the code electromagnets as a function of the content of the information to be recorded and the position of the constantly rotating character drum. Information for printing a single line is received in 1-2-4-8 parallel binary-coded decimal code through 64 channels. The unit operates with the information source in the request-response mode. Each code pulse from the source must be accompanied by a "Request" signal and information can be recorded in the buffer storage only with the presence of a "Readiness" signal from the unit. The possibility of an independent check of the unit's working ability is provided for in it in the "Test" mode.

Specifications of Shch68000K Unit

Maximum printing speed in lines per second	30
Number of:	
Digits in a line	16
Printed characters	16
Printed characters	0 to 9, + , point, - , % , space, *
Size of printed characters in mm	2.1 X 1.4
Width of paper tape in mm	45
Parameters of coded input signals:	
Level of logical "0" of positive and negative polarity in V, not greater than	0.6
Level of logical "1" of positive and negative polarity in V	3 ± 0.6, 6 ± 1.2, 12 ± ± 2.4, 24 ± 2.4
Supply voltage in V	220 +10, -15 percent (50 Hz ± 1 percent)
Power requirement in W	130
Overall dimensions in mm	130 X 480 X 485
Weight in kg	18

Orders for the unit can be sent to the following address: 172218, Moscow, ul. Krzhizhanovskogo, 16, Telex 112906.

Information can be obtained from the following address: 357036, Nevinnomyssk, 6, Stavropol'skogo kraya, ul. Gagarina, 217, Nevinnomysskiy zavod elektroizmeritel'nykh priborov [Nevinnomyssk Electronic Measuring Instrument Plant].

8831
CSO: 1863/80

UDC 681.325.21

GRAPHIC DATA INPUT UNIT

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 pp 7-8

[Article by A.D. Bekh, V.V. Chernetskiy and A.P. Ganin, engineers]

[Excerpts] Characteristic of the state of the art of the development of graphic data input units is the enormous diversity of design principles and engineering solutions. Unique models of mechanical, acoustic, piezoelectric, optical, electrical and electromagnetic units have been created. However, in connection with the noted extensive introduction of microprocessors into the national economy, quite urgent is the task of creating input units which will not be inferior to microprocessor equipment in terms of availability and technical and economic parameters.

This problem has been solved at the Ukrainian SSR Academy of Sciences Institute of Cybernetics (Kiev) by studying the possibilities of using the physical and technological base of microelectronics for designing these units. An array of discrete coding elements based on strip transformers fabricated by the photochemical printing method is used as the working field of the plotting board and standard logic microcircuits are used as the electronics for controlling the plotting board's working field. The coordinate mapper has single-turn coils also produced by the printed method on a flexible foil-covered insulator.

The graphic data input unit developed differs advantageously from the best foreign models in terms of overall size, weight and power requirement with a smaller digitization interval of the working field (cf. comparative specifications in table 1).

Table 1.

Characteristic	Unit developed	Quest Automation firm's unit
Working field of plotting board in mm	320 X 320	1290 X 914
Number of coordinate lines, units	256 X 256	256 X 182
Maximum thickness of data medium in mm	0.4	0.4

[Continued on following page]

Rate of output of coordinates in Hz	2 to 2000	0.6
Power requirement from network in V·A	35	360
Overall dimensions in mm	640 X 540 X 60	2286 X 177 X 1828
Weight in kg	8	226

The 320 X 320 mm working field contains 256 single coordinate lines located in directions x and y and 1.25 mm apart from one another. It is produced by the method of photochemical etching of a foil-covered insulator, usually used in multilayer printed circuitry.

The 2-stage current switch is constructed from series K155 elements. The amplitude of the current pulse in coordinate lines is 40 mA with a duration of 100 ns and they are formed by type K155LA7 microcircuits.

The presence of zones of sensitivity on the working field of the plotting board makes it possible to output sound signals to the operator when the coordinate mapper is set between coordinate lines and at the same time to eliminate errors in coding digital data. On the other hand, the zones of sensitivity prevent the use of the working field as a unit for determining the coordinates of random points. The parameters of a measuring unit with a resolution of 1.25 mm are realized by lowering the triggering level of the threshold element by the amount by which the zones of sensitivity of neighboring coordinate lines overlap.

The unit is powered from an a.c. line by means of a built-in transformerless stabilized d.c. source.

The specifications of the unit developed and of the widely used digital coder from the English Quest Automation firm are presented in table 1.

The small overall size and low weight and power requirement make it possible to use the unit at the ordinary work place of a designer.

8831
CSO: 1863/80

UDC 621.398:654.94

DATA DISPLAY UNIT WITH DYNAMIC INDICATION

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 pp 25-26

[Article by A.M. Gladkov and N.V. Nagayets, engineers, and V.A. Teslenko, candidate of technical sciences]

[Excerpts] The trend toward improving the accuracy and resolution of digital measuring instruments has resulted in an ever increasing number of digits to be indicated on a readout unit. Taking into account indication of additional information on the sign, units of the quantity measured and the power, the number of digits to be indicated in modern high-precision measuring instruments can equal 10 or more.

A display has been proposed for microprocessor instruments which consists (fig 1) of a pulse generator, G; a binary pulse counter, SI; anode and cathode decoders, DA and DK, respectively; a RAM; a code switching circuit, SPK; and a gas-discharge element display panel, IP. Indication is accomplished in the following manner. Pulses from the generator's output enter the binary pulse counter, from which a code is sent to the anode decoder which successively switches the anodes to the anode voltage source. The counter's output code through the code switching circuit also enters the address input of the RAM, where the information to be displayed is recorded. From the output of the RAM the information is sent to the cathode decoder, which converts the binary-coded decimal code into 7-segment code.

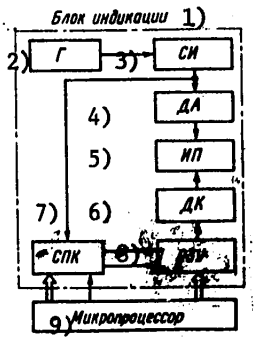


Figure 1.

Key:
1. Display unit
2. Pulse generator
3. Pulse counter
4. Anode decoder
5. Display panel
6. Cathode decoder
7. Code switching circuit
8. RAM
9. Microprocessor

At a specific moment of time anode voltage is established at one of the anodes of the display panel. The information for the lamp (digit) in question is read out from the RAM. With the arrival of a new pulse from the generator the code in the pulse counter is changed and the next digit of the display panel is connected, and information of the next line is read out from the RAM. Changing of information in the RAM is accomplished as follows. The code of the RAM line (digit) in which information is to be changed in the code switching circuit enters from the address output of the microprocessor. Upon the "Write" instruction from the microprocessor this circuit cuts off the address inputs of the RAM from the pulse counter and connects them to the address outputs of the microprocessor. Data from the information output of the microprocessor are entered into the RAM according to this address. The entry of information into the following line takes place similarly.

When necessary, specific digits of the display panel are cleared by entering a forbidden code into the RAM.

Gas-discharge display panels of the GIP 11 or IGP-17 type and a RAM of the K155RU2 type are used in the display. The maximum number of digits which can be displayed is 16.

The flowchart of the program for controlling the proposed display by means of a K580IK80 microprocessor is presented in fig 2, and a program in the ASSEMBLER language can be obtained at Kiev Polytechnical Institute, where this unit was developed.

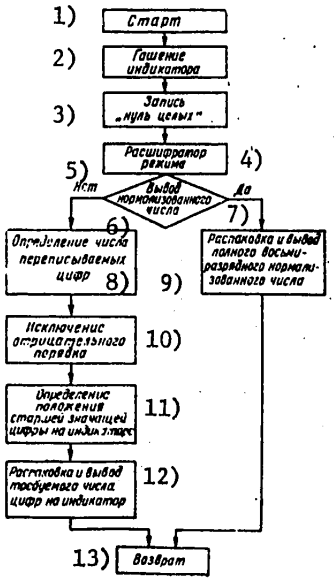


Figure 2.

Key:

1. Start
2. Clear display
3. Write "zero integers"
4. Mode decoder
5. No
6. Output of normalized number
7. Yes
8. Determination of number of digits to be rewritten
9. Unpacking and output of complete 8-digit normalized number
10. Exclusion of negative power
11. Determination of position of highest-order digit on display
12. Unpacking and output of required number of digits to display
13. Return

8831
CSO: 1863/80

GRAPHIC DISPLAY UNITS

UDC 681.3

'GRAFIT' GRAPHIC DISPLAY

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 pp 21-22

[Article by L.M. Arsent'yev, V.A. Krupnova, A.P. Fedoseyev and V.M. Khmyachin, engineers]

[Text] Automated design systems have been based to an ever greater degree on automated work site (ARM) complexes [1, 2]. However, know-how gained from using them has demonstrated that the technical characteristics of the graphic displays included in ARM-R and ARM-M complexes are unsatisfactory for solving specific design problems. Therefore at the present time work is under way on improving and modernizing them.

In this article the results are given of work relating to the creation of the "Grafit" graphic display designed to replace displays in the ARM-R (EPG-400) and ARM-M (UPGI-A) complexes.

It is obvious from table 1 that the "Grafit" considerably surpasses EPG-400 and UPGI-A displays in the size of the working field of the screen, the number of addressable points and the information capacity of the screen and its functional capabilities have also been expanded in the area of the structural organization of graphic information, input/output and the organization of graphic programming.

Table 1. Comparative Characterization of EPG-400, UPGI-A and "Grafit" Displays

Characteristic	EPG-400	UPGI-A	"Grafit"
Image elements formed	Vector, character	Vector, character, arc, circle	Vector, point, character, arc, circle
Structural organization of graphic information	-	Element, unit, array	Element, unit, array, standard element, subroutine
Input	Keyboard, computer	Photoelectric reader, computer, keyboard, "Consul" typewriter	Photoelectric reader, computer, keyboard, coding plotting board, "Consul" typewriter

[Continued on following page]

Output	Computer	"Consul" typewriter, punch, computer	"Consul" typewriter, punch, computer
Image transformation	-	Shift, rotation, scaling, symmetry, change of types of lines, editing of text	
Identification	Indication by means of light pen		Indication by means of light pen and by name from functional keyboard
Information capacity of screen:			
Time for drawing a vector 240 mm long, in μs	30	25	25
Number of circles	-	160	180
Number of characters	2000	800	2000
Size of working field in mm	200 X 200	297 X 210	420 X 340
Number of addressable points	1024 X 1024	1024 X 1024	2048 X 2048
Size of plotting board in mm	-	-	460 X 380

The external appearance of the "Grafit" display is shown in fig 1. Its structure and software are discussed below.

The display is designed according to the principle of a multilevel hierarchical structure (fig 2). Each level is an element functionally and structurally complete in keeping with the hierarchy. The bottom level is a set of function generators (vectors, circles, characters) and a light pen module. The function generators are designed to convert the digital codes of graphic elements into control voltages of the CRT's deflection system.

The next level is the display controller level. The display controller is a special-purpose high-speed processor with a clock rate of 8 MHz. It performs the functions of an electronic switch for display file formats between function generators, controls the restoration of the display file, and also processes signals arriving from the light pen module.

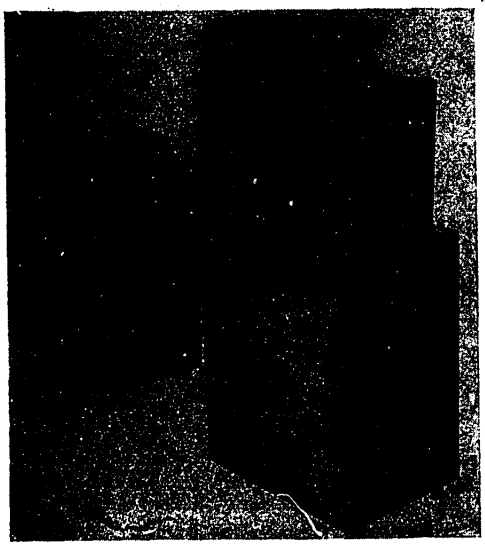


Figure 1. External Appearance of "Grafit" Display

The display file is formed at the level of interpretation of the input language. This level includes an "Elektronika-60" microcomputer, a 16K-byte basic subroutine RAM (ROM), a coding plotting board and a keyboard.

The last level--the graphic programming level--is designed for the compilation of graphic programs and for offering users opportunities to orient the display to problems of specific classes. In terms of hardware the graphic programming level consists of an "Elektronika-60" microcomputer with a 24K-byte RAM, punched tape input/output units and a "Consul" typewriter. Coupling between levels is accomplished according to the "common line" standard. Interfacing of the "Grafit" display with the ARM central processor is provided for in keeping with the same standard. The display's software, just as the hardware, is constructed according to the multilevel hierarchical principle. Each level is defined by its own language and functions.

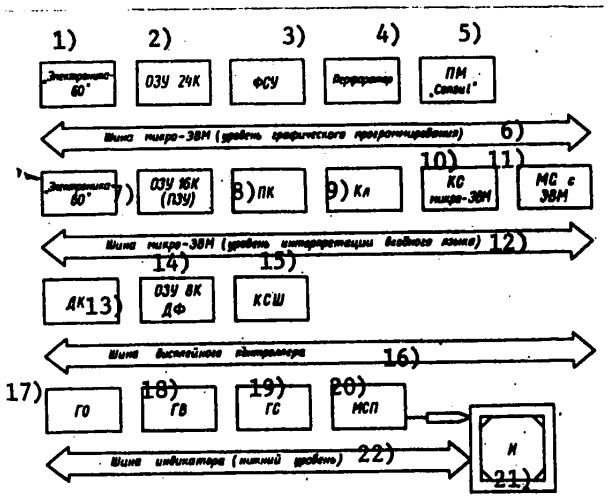


Figure 2. Structure of "Grafit" Graphic Display: PK--coding plotting board; Kl--keyboard; KS--interface system; MS--interface module; GK--display controller; DF--display file; KSSh--line interface system; GO--circle generator; GV--vector generator; GS--character generator; MSP--light pen module; I--display

Key:

1. "Elektronika-60"
2. 24K RAM
3. Photoelectric reader
4. Punch
5. "Consul" typewriter
6. Microcomputer line (graphic programming level)
7. 16K RAM (ROM)
8. PK
9. Kl
10. Microcomputer KS
11. Computer interface module (MS)
12. Microcomputer line (input language interpretation level)
13. DK
14. 8K RAM, DF
15. KSSh
16. Display controller line
17. GO
18. GV
19. GS
20. MSP
21. I
22. Display line (bottom level)

The highest level is represented by the graphic programming language. A compiler for this language accomplishes its compilation into the "Grafit's" input language. The input language is converted into the formats and array of the display file at the input language interpretation level. The display file, in turn, is translated by means of microprograms of the display controller into the formats of function generators. Let us briefly describe individual groups of input language statements.

Mode Statements

These statements load the software from punched tape, diagnose the state and set the operating mode of the display. The "Grafit" can operate in four modes: automatic, "Keyboard," "Computer" and "Editing." In the automatic mode the display functions according to a program written previously at the graphic programming level. Operation with a function keyboard, light pen and coding plotting board magnetic pen is accomplished in the "Keyboard" mode.

Setting of the "Computer" mode converts the "Grafit" into a computer terminal and it operates through channel instructions supplied from the computer to which the display is connected.

Only keys for editing textual information are active in the "Editing" mode, i.e., keys for the replacement, insertion and removal of characters and for indicating the sequence for printing out lines of text. In the "Editing" mode the tag can be moved along lines of text from character to character by means of appropriate statements.

Graphic statements together with a group of "Attributes" statements are designed for forming the elements of a sketch, such as points, segments, arcs and characters with the required types of lines and orientation and height of characters.

Structural Organization Statements

These determine the structural units of graphic information, i.e., units and standard elements. The latter represent combinations of elements of the sketch named which have been entered into a library of standard elements, from which they can be called for review or for placement at required points on the screen.

Units represent the next structural unit of the display's input language. They include standard elements and sketch elements (graphic statements). Units can be formed from graphic elements or from units formed earlier.

Shift, rotation, symmetry, vector division, measuring and erasure statements form the group of transformation statements. The arguments of transformation statements are structural units of the input language labeled by means of the group of identification statements: graphic elements or units of the entire array of graphic information stored in the display's memory.

Input/Output Statements

These accomplish the input/output onto punched tape and a typewriter of graphic and textual information. Several formats are provided for the representation of information on media: display files, symbolic representation of graphic information, textual, etc.

Service Statements

This group includes statements which perform auxiliary functions. They include initial setting and change of keyboard statements and also an "Arithmetic"

statement by means of which it is possible to compute arithmetic expressions and elementary functions.

Bibliography

1. Presnukhin, V.V. and Yelshin, Yu.M. "Automated Work Site for a Developer of Radio Electronic Equipment," OBMEN OPYTOM V RADIOPROMYSHLENNOSTI, No 12, 1976.

2. Nazarov, V.V. and Fedoseyev, A.P. "Hardware Complex and Software of an Automated Work Site for a Developer of Machine Building Products," PRIBORY I SISTEMY UPRAVLENIYA, No 1, 1979.

8831
CSO: 1863/80

UDC 681.327

CHARACTER-GRAPHIC INFORMATION DISPLAY

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 pp 23-24

[Article by A.N. Shesterkin, A.M. Smolyarov, B.I. Levkoyev, candidates of technical sciences, and V.N. Sharov and G.F. Tyulenev, engineers]

[Excerpts] Information displays make possible the efficient interaction between a human being and automatic equipment. The use in them of gas-discharge display panels (GIP's) makes it possible to construct in a number of cases quite ideal displays [1, 2]. Units for the reproduction of graphic [3-5] and character [6-7] information with GIP's are well known. Character-graphic displays are the most efficient.

The design principles of these units are well known, but information on their development is practically absent in the published data. The results of designing and studying a character-graphic display executed with an alternating-current GIP are discussed in this article.

The structural diagram of a character-graphic display developed at the Ryazan' Radio Engineering Institute is shown in fig 1.

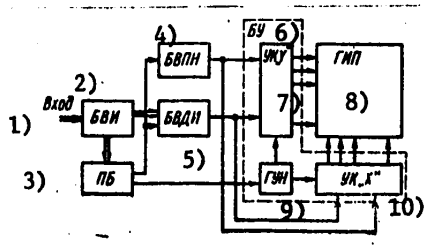


Figure 1.

Key:

1. Input
2. Input information block
3. Program block
4. Conditionally constant information reproduction block
5. Dynamic information reproduction block
6. Control block
7. "Y" switch
8. Gas-discharge display panel
9. Control voltage generator
10. "X" switch

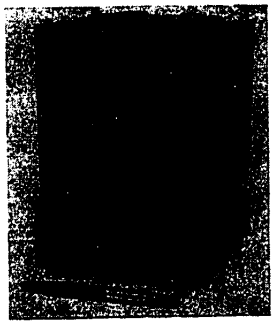


Figure 2.

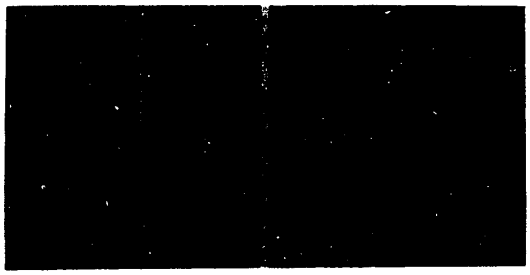


Figure 3.

The sustaining voltage time is 20 μs. A 14-fold repetition of control pulses is employed for the purpose of improving the reliability of the lighting and extinction of cells. The length of a single cycle for writing (erasing) information on the panel is 300 μs.

The circuit diagram of the display is executed with series 133, 134 and 556 integrated elements. The second stages of the "X" and "Y" switches are constructed with 1NT251 and 2TS622 transistors and 2D906 diodes without using transformers, and the sustaining voltage generator with 2T809 and 2T630 transistors and TII5-3 transformers. The control block (fig 2) has a low power requirement (less than or equal to 30 W) and guarantees the sustaining and lighting (extinction) of any cell of the panel with a variation in supply voltage over the range of ± 3 percent and of the ambient temperature from -40 to +70 °C.

Individual frames of images produced by means of the unit developed are shown in fig 3. The display has been designed by using a panel with a number of cells of 128 X 128. The use of panels with a greater number of cells will make it possible to improve considerably the accuracy and quality of the display of information.

Bibliography

1. Kashnikov, N.G., Pokryvaylo, A.B. and Yablonskiy, F.M. "Gas-Discharge Display Panels, a New Class of Instruments for Displaying Data," ELEKTRONNAYA TEKHNIKA. SER. 4. ELEKTROVAKUUMNYYE I GAZORAZRYADNYYE PANELI, No 8, 1976.

2. Yablonskiy, F.M. "Gazorazryadnyye pribory dlya otobrazheniya informatsii" [Gas-Discharge Devices for Displaying Information], Moscow, Energiya, 1979.

3. USSR Patent No 636644.

4. Smolyarov, A.M., Borzykh, V.Ye., Voskresenskiy, A.V. et al. "Graph Display Employing a Gas-Discharge Array Panel," PRIBORY I SISTEMY UPRAVLENIYA, No 7, 1977.

5. Berkutov, A.M. and Proshin, Ye.M. "Matrix Display Panels in Digital Oscillography," ELEKTRONNAYA TEKHNIKA. SER. 4. ELEKTROVAKUUMNYYE I GAZORAZRYADNYYE PANELI, No 8, 1976.

6. Lavrent'yev, S.I., Sviyazov, A.A. and Smolyarov, A.M. "Unit for Displaying Character Information on Gas-Discharge Display Panels," PRIBORY I SISTEMY UPRAVLENIYA, No 12, 1979.

7. Danilov, V.G. and Melikhov, G.I. "Reproduction of Character Information on a Flat Screen of the Plasma Type," ELEKTRONNAYA TEKHNIKA. SER. 3. GAZORAZRYADNYYE PRIBORY, No 2, 1970.

8831
CSO: 1863/80

UDC 621.327.12

MINIATURE INFORMATION DISPLAY UNIT UTILIZING IMG-1 DISPLAY

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 pp 24-25

[Article by S.I. Lavrent'yev, V.K. Kalmykov, engineers, and A.M. Smolyarov, candidate of technical sciences]

[Excerpts] Units employing direct-current gas-discharge display panels (GIP's) come under the heading of promising information displays for individual use. In this article is discussed a character information display employing an IMG-1 matrix-type gas-discharge display developed at the Ryazan' Radio Engineering Institute [1] and distinguished by the familiar one in [2] by a simpler control circuit and smaller size and lower weight.

Characters are formed by means of a 5 X 7-cell microraster, which makes it possible to obtain the usual representation. Characters of the Russian and Roman alphabets, Arabic numerals and special characters are formed in the unit. Information can be input into the unit from an input console, a "Consul" electric typewriter or from the output of a computer. The write mode has priority over the display mode, i.e., when information is entered the display stops until writing is finished. Series K133, K565 and K505 integrated microcircuits serve as the element base of the unit. The hardware cost for the display of a single character with this element base is comparable to the cost for lamps of the IN type and is considerably lower than with light-emitting diode displays of this size. It should be mentioned that 70 percent of the cost of purchased items is due to the IMG-1 display, whose cost has a tendency to drop. The power supply is designed with a d.c. voltage transformer, which makes it possible to operate both from a 220 VAC (50 Hz) source and from a +24 VDC source. The external appearance of the unit is shown in fig 2.

Specifications of Unit

Number of character positions	160
Height of character in mm	7
Color of light	Orange-red
Brightness in cd/mm^2	50
Operating temperature range in °C	From -10 to +45
Power requirement in V•A	25
Overall dimensions in mm	145 X 160 X 180
Weight in kg	2.5

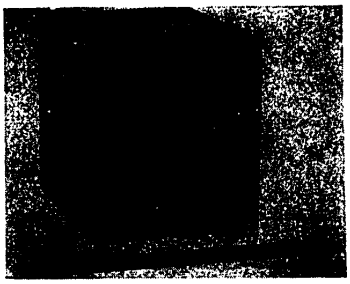


Figure 2.

This unit can find an application in automated control, testing and communications systems, as well as as a character information display in control computers and other equipment.

Bibliography

1. Glubokov, G.G. and Loban, V.I. "Matrix Gas-Discharge Display Utilizing 10^4 Display Elements, Controlled by Bipolar Microcircuits," ELEKTRONNAYA TEKHNIKA. SER. 4. ELEKTROVAKUUMNYYE I GAZORAZRYADNYYE PRIBORY, No 8, 1976.

2. Lavrent'yev, S.I., Sviyazov, A.A. and Smolyarov, A.M. "Unit for Display of Character Information Employing Gas-Discharge Display Panels," PRIBORY I SISTEMY UPRAVLENIYA, No 12, 1979.

8831
CSO: 1863/80

UDC 681.327

COLOR CHARACTER-GRAPHIC DISPLAY

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 p 25

[Article by I.I. Zhokhov, engineer]

[Text] This display is designed for use under laboratory conditions and for displaying information in the form of graphs, histograms or an alphanumeric text on the screen of a color video monitor (VKU). The input information is a digital code arriving from digital measuring equipment or a computer. The display makes possible editing of the information displayed, automatic color tolerance testing, the output of information to a punch for purposes of documentation, the formation of information in steps of 10 relative units of the reference scale, and optional numbering (according to the operator's wish) of any ordinate of one of six graphs or histograms.

The video monitor's screen measures 300 X 266 mm. Colors for the display of graphic information are red, purple, blue, green, yellow and light blue, and for textual information, green. The maximum number of graphs which can be displayed simultaneously is six and of histograms one, of graphic information abscissas 64 and of ordinates $\pm$ 127, and of lines of text 12 of 32 characters each: Russian and 7 Roman characters, digits and special characters in keeping with GOST [All-Union State Standard] 13052-74.

Key Specifications

Supply voltage in V	220 + 10, -15 percent (50 Hz)
Power requirement in V•A	650
Overall dimensions in mm	850 X 1000 X 1300
Weight in kg	200

The levels of the display's input and output signals are consistent with series 155 TTL logic.

Interfacing of the display with external information sources is accomplished according to OST [Industry Standard] 25.190-73, "GSP [State System of Industrial Instruments and Automation Equipment]. Modular Complexes of Instruments and Automation Equipment. YeI1 Interface."

The display is connected to information sources having interfaces different from the one indicated via additional peripheral devices not produced by the plant. The external appearance of the display is shown in fig 1.

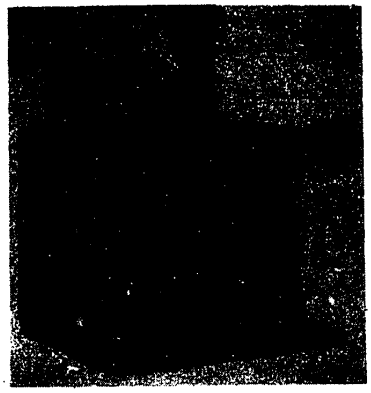


Figure 1.

The N712 display is produced in two modifications: with a video monitor and without one. Information can be gotten by writing to the following address: 357036, Nevinnomyssk-6, Stavropol'skogo kraya, ul. Gagarina, 217, Nevinnomysskiy zavod elektroizmeritel'nykh priborov [Nevinnomyssk Electronic Measuring Instrument Plant].

8831
CSO: 1863/80

SOFTWARE

ABSTRACTS FROM JOURNAL 'PROGRAMMING', NOVEMBER-DECEMBER 1981

Moscow PROGRAMMIROVANIYE in Russian No 6, Nov-Dec 81 pp 94-96

UDC 518.74: 007: 57

LOGIC AND ALGORITHMIC FORMALISM FOR PROBLEM OF WRITING CORRECT PROGRAMS

[Abstract of article by Nepeyvoda, N.N.]

[Text] Formalization of proof and semantics in the logic of predicates oriented to the problem of writing programs is considered. Logic synthesis is illustrated through examples.

UDC 681.3.519

PARSING ALGORITHM FOR CONTEXT-DEPENDENT GRAMMARS

[Abstract of article by Vol'dman, G.Sh.]

[Text] A parsing algorithm is suggested for nondegraded grammars; it is a generalization of the Early algorithm. The correctness of the algorithm is proved. The operation of the algorithm is illustrated using the example of a disseminated grammar.

UDC 681.3.06

METHOD FOR OPTIMIZING LR-ANALYZERS

[Abstract of article by Surkova, L.V.]

[Text] A class of RSA-grammars is introduced whose determination is based on break-down of the KS-grammar alphabet into nonintersecting position classes. A method is described for constructing the determinate LR-analyzer for the languages of this class.

UDC 681.3.06.51

MODEL APPROACH TO WRITING FAMILIES OF APPLIED PROGRAM PACKAGES

[Abstract of article by Parasyuk, I.N. and Sergiyenko, I.V.]

[Text] An approach based on a developed principle of modularity is proposed for writing families of applied program packages. Basic program components realizing this approach are examined.

UDC 681.3.06

NOTATION FOR CONVENTIONAL EXPRESSIONS FOR STRUCTURAL PRESENTATION OF DATA AND PROCEDURES

[Abstract of article by Kulaichev, A.P.]

[Text] In this article a linear-continuous notation is suggested for conventional expressions as a unified means for brief description of data structures and procedure structures; the expressions do not use the concept of labels or transfer operator.

UDC 681.3.068

APPLICATION OF PREPROCESSOR FACILITIES IN FORTRAN AND ALGOL

[Abstract of article by Bezborodov, Yu.M. and Lavrent'yeva, T.G.]

[Text] Questions of using preprocessor facilities of PL/1 in the operating system of the YeS computer when programming problems in FORTRAN and ALGOL are considered. A standard procedure is described for writing programs in these languages.

UDC 681.3.06: 800.92

RELATIONAL LANGUAGE IN PROGRAMMING AND PRINCIPLES FOR REALIZING IT ON A SERIAL COMPUTER

[Abstract of article by Kleshchev, A.S.]

[Text] (Relyap) parallel programming language is based on a relational computing model. Its syntax and contextual conditions guarantee correctness of computations. The possibilities of double determination of computations are discussed.

UDC 681.3.06

LIST PROCESSING IN THE YeS COMPUTER

[Abstract of article by Khmel'nik, S.I.]

[Text] A LISP system is described; it is realized on a YeS computer and is distinguished by its additional possibilities for describing and processing list structures, simplification of certain language constructions, and the methods for realization of the interpreter.

UDC 681.3.06.51

ALGORITHM LANGUAGES FOR SPECIALIZED MINICOMPUTERS AND MICROPROCESSORS

[Abstract of article by Serebrovskiy, L.A. and Lipayev, V.V.]

[Text] An analysis is made of the requirements made of software for small specialized computers. A system is proposed for languages mutually linked in regard to structure and display facilities used for programming at various levels. The need is formulated for language systems for selecting the high level language reference.

UDC 681.142.1

A METHOD FOR RATIONAL GROUPING OF OPERATIONS

[Abstract of article by Novikov, Ye.S.]

[Text] The problem of breaking down software for a computing system into linked groups of operations is considered with respect to the criterion for minimum time spent on comprehensive debugging of programs. A method is described for selecting rational characteristics for these groups of operations. A constructive algorithm is shown for grouping of operations.

UDC 518.74: 007: 57

DEDUCTIVE SEMANTICS FOR THE 'EXECUTE' OPERATOR IN COBOL

[Abstract of article by Kasatkina, I.V. and Petrushin, V.A.]

[Text] Semantics of the EXECUTE operator in COBOL are described axiomatically. Output rules are shown for operator formats realizing call procedures and formation of cyclic structures. Using as an example the proof for the correctness of COBOL programs, the application of the proposed axiomatics is shown for verification programs for data processing.

UDC 681.3

REALIZATION OF PROGRAM INTERFACE FOR THE MIR-1 and YeS COMPUTERS

[Abstract of article by Vaganov, S.A. and Kolyakin, Yu.D.]

[Text] Characteristics are shown for a MIR-FORTRAN IV (YeS) converter system which is a system for translating programs in ALMIR-65 for the MIR-1 computer into programs in FORTRAN IV for the YeS computer disk operating system.

UDC 681.3.001

COMBINED FORMAT FOR OUTPUTTING NUMERICAL DATA FOR DESIGN DOCUMENTS

[Abstract of article by Motyl', D.N., Sokolinskiy, Yu.A. and Farber, K.S.]

[Text] A combined format is described for presentation of numerical data that makes it possible to output numbers with a given relative error. The format described is realized in a PL/1 function procedure in the operating system of the YeS operating system.

9642
CSO: 1863/76

INTRODUCTION TO THEORY OF ANALYSIS AND PARALLEL OPERATION OF COMPUTER PROGRAMS DURING TRANSLATION

Moscow VVEDENIYE V TEORIYU ANALIZA I RASPARALLELIVANIYA PROGRAMM EVM V PROTSESSE TRANSLYATSII in Russian 1981 (signed to press 9 Jul 81) pp 2-4, 254-255

[Annotation, foreword and table of contents from book "Introduction to Theory of Analysis and Parallel Operation of Computer Programs During Translation", by Eduard Anatol'yevich Trakhtengerts, Izdatel'stvo "Nauka", 3250 copies, 256 pages]

[Text]

Annotation

This book is devoted to syntactic analysis of input texts and parallel programming methods for the computational process. The book examines properties of various grammars which generate programming languages, sequential and parallel syntactic analysis algorithms based on these properties and methods for optimizing them. Methods for parallel programming and algorithms for parallel calculation of arithmetic expressions and systems of equations are examined.

The book is intended for scientific and engineering-technical workers in the area of programming and software.

Foreword

Regardless of the fact that intensive work on programming theory is underway, there are many areas of programming in which no theoretical results have been obtained which allow the problem of program construction to be formalized and to use that formalization as the basis for developing and substantiating solution methods.

One of the areas in which theoretical results have been used extensively in practice is the development of translation systems. One of the first problems to be solved in translation theory was that of sequential syntactic analysis. Based on the methods of formal description of programming languages, it was not only possible to create a mathematical theory of sequential syntactic analysis, but to use it in practice as well. As a result, the amount of work involved in developing syntactic analyzers has been reduced by factors of ten. Methods based on the theory of sequential syntactic analysis have already become traditional in programming practice.

In connection with the intensive development of multiprocessor computing systems, programmers have relatively recently been faced with the problem of parallel programming which, in turn, has imposed requirements of its own on the construction of translators. It has been necessary to develop parallel translation processes, as well as parallel operation of programs obtained as the result of translation. In this connection, theoretical research has recently been done on parallel translation; this research has provided the basis for creating methods of parallel syntactic analysis which are a natural development of traditional sequential methods. Active work on parallel computing theory has begun. As a result, methods have been developed for parallel output programming during the translation process, algorithms have been created for parallel computation of some types of arithmetic expressions and systems of equations and some estimate of parallel programming efficiency have been obtained. However, methods based on results from the theory of parallel computation are still underutilized in practice.

This monograph is devoted to two problems in translation theory: syntactic analysis and organization of parallel computation during the translation process. The book consists of two parts, with the first investigating sequential syntactic analysis method and the second examining parallel syntactic analysis and parallel programming during the translation process.

The book presents the theory of the matter, and where possible also presents algorithms which have been used in practice.

Table of Contents

Foreword 1

I
SEQUENTIAL SYNTACTIC ANALYSIS

Chapter 1
Some Concepts and Definitions 5

§ 1.1. Sets 5
§ 1.2. Formal grammars and languages 7
§ 1.3. Problems in syntactic analysis 15

Chapter 2
Precedence Relationship Grammars 20

§ 2.1. Informal introduction 20
§ 2.2. Context-sensitive relationship and precedence function grammars 27
§ 2.3. Equivalent transformation of context-sensitive grammars to precedence grammars 36

Chapter 3
LR(k)- and LL(k)-Grammars 46

§ 3.1. Informal introduction 46
§ 3.2. Some definitions and algorithms 61
§ 3.3. Optimization of control table 69
§ 3.4. De Remer's algorithm 88
§ 3.5. LL(k)-grammars 90

Chapter 4
Expansion of LR(k)-Grammars 99

§ 4.1. G(k)-grammars 99
§ 4.2. ELR(k)-grammars 109

II
ANALYSIS AND PARALLEL PROGRAMMING IN MULTIPROCESSOR COMPUTING SYSTEMS

Chapter 5
Efficiency Estimates of Parallel Computation 117

§ 5.1. Some definitions 117
§ 5.2. Systems with single instruction stream. Mainline data processing 120
§ 5.3. Systems with single and multiple instruction stream. Parallel data processing 123

Chapter 6
Parallel Syntactic Analysis 130

§ 6.1. Parallel-sequential analysis algorithm using matrices and precedence functions 130
§ 6.2. Equivalent transformation of context-sensitive grammars to grammars with four precedence relationships 149
§ 6.3. Equivalent transformation of context-sensitive grammars to form having precedence functions 159
§ 6.4. Two-way analysis algorithm based on inverse-recursive parametric grammars 170

Chapter 7
Analysis of Programs for Automatic Parallel Programming 174

§ 7.1. Possible stages of parallel programming 174
§ 7.2. Partitioning programs into weakly-connected sections 176
§ 7.3. Finding parallelism between operators within linear sections 184

Chapter 8
Transformation of Arithmetic Expressions and Equations for Parallel Computation 188

§ 8.1. Transformation of arithmetic expression of general type 188
§ 8.2. Polynomial transformation 195
§ 8.3. Transformation of linear recursive expression 200
§ 8.4. Transformation of general linear recursive system of n equations 205

Chapter 9
Parallel Programming of Cycles During Translation Process 218

§ 9.1. Parallel programming of cycles using hyperplane method 218
§ 9.2. Parallel programming of cycles using coordinate method 241

Bibliography 250

6900
CSO: 1863/57

UDC 681.3:519.14

USING PARALLEL CALCULATIONS MODEL TO INVESTIGATE STRUCTURE OF ORIENTED GRAPHS

Moscow MIKROELEKTRONIKA in Russian Vol 10, No 4, Jul-Aug 81 (manuscript received 21 Jan 80) pp 302-307.

[Article by Ya. A. Grundspen'kis, Riga Polytechnical Institute]

[Text] Graph theory, being a simple and descriptive tool for solving a number of digital mathmmatics problems, has recently found ever-wider application in various fields of science and technology. The use of methods and algorithms of graph theory has made it possible to formalize a number of problems of diagnostics [1], structural analysis of complex systems and determination of typical structural configurations in unit systems [2] and has also made it possible in principle to solve such difficult problems as, for example, identification of graph structures and the "travelling salesman problem." Main attention is devoted in all these problems to investigating the structure of a real system, i.e., to determination of the direct and indirect relationships between the vertices of the corresponding graph. All the varieties of the general problem of investigating the structure of graphs--finding the set of shortest paths between any pair of vertices, determining the set of all paths or oriented cycles (orcycles), determination of strongly related components and so on--are laborious and require large expenditures of machine time and computer memory. Therefore, the efforts of developers of the methods and algorithms used in practice, in which sequential calculation procedures are employed, are directed toward theoretical substantiation of the possibility of finding algorithms, the number of operations of which would be linearly dependent on the number of vertices and arcs of the graph.

The purpose of the given paper is to consider and evaluate the prospects of utilizing the parallel calculations model to investigate the connectedness of oriented graphs (orgraphs).

Following [3] and the terminology of [4], let us consider parallel calculations in graph theory in two aspects: 1) when the investigator has at his disposal an infinite number of identical parallel processors (the so-called unlimited case) and 2) when the number of processors is limited. It is assumed in this case that each processor can perform binary arithmetic and comparison operations per unit time (during one working step); the input data are entered in the memory prior to the beginning of calc[illegible]ons, each processor takes operands from the memory and

enters the result in the memory after each step and only one processor can change the contents of any section of memory at each step. Some deficiencies of this model will be shown below but first let us consider the algorithm for determining all the nonrepeating orcycles using parallel calculations.

The algorithm for finding orcycles that actually utilizes the idea of parallel calculations is presented in [5]. The structure of this algorithm clearly shows that the "wide search" procedure is realized in it, i.e., all the arcs emerging from it are considered simultaneously upon emergence to each new vertex. Since only one processor was used in the given case, a shift of some set of paths in the set already found was required so as to write new paths in the computer memory. This undoubtedly reduced the advantages of the developed algorithm but its practical application showed that it was slightly inferior to algorithms known at that time that realized the "deep search" procedure [6, 7], thus seemingly confirming the potential capabilities of parallel calculations.

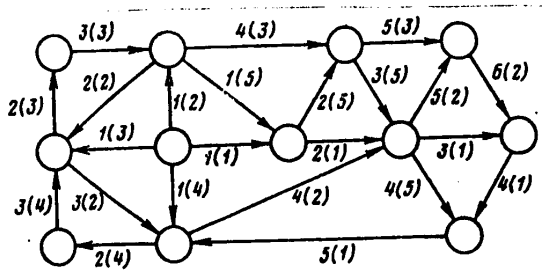


Figure 1. Parallel Calculations in Orgraph. The numbers denote the order of passage through arcs and the numbers of processors are indicated in parentheses

Let us consider a diagram of parallel calculations for the orgraph presented in Figure 1. It is obvious that each process builds only a partial list (the n-ratio in the graph) which contains "new" (untraversed) arcs and all the found partial lists must be combined at the end of calculations. The following conclusions follow from the example.

1) An algorithm constructed on the considered principle is more effective, the more the orgraph is "filled," i.e., the higher the ratio m/n (where n is the number of vertices and m is the number of arcs). We note that a similar conclusion is made in [3] with respect to this algorithm that only with an adequately "filled" graph are some parallel procedures close to optimum in the sense that the time expenditures for them require $S/K + \varepsilon$ steps (where S is the number of steps of the best algorithm that utilizes the "deep search" procedure and ε is a small value). However, additional experimental and theoretical confirmations are required for these conclusions.

2) If a limited number of processors participates in the parallel procedure, then main attention should be devoted to the "load" of processors, i.e., all processors should be involved in the search for new arcs. If they are not (the case of rare graphs), the processors must be set in agreement to the vertices and thus one must

consider several vertices simultaneously, which is not provided in the model of [3].

3) Unlike the "wide search" procedure [6], where the concept of a graph in the form of a list is more advantageous for computer entry, the input data for parallel calculations can be represented by a matrix of contiguities, since there is no need to expend extra steps in this case to establish the fact that a specific arc exists. Moreover, parallel calculations increase the capability of those algorithms which are based on the use of sequential raising of the matrix of contiguities to a power [7].

To estimate the algorithms based on parallel calculations, one must at least approximately find the lower and upper bounds of the number of required steps. The theorem of [7] may be used to estimate the lower bound in the case when the graph is represented by a matrix of contiguities.

Let there be required $R \geq 1$ binary arithmetic operations to calculate the unit value of R used. The shortest K-calculation of R then occupies at least $\{[(r + 1) - -2^{\log_2 K}]/K\} + (\log_2 K)$ steps if $r \geq 2^{\log_2 K}$, and $[\log_2 (r + 1)]$ steps in the opposite case. In both cases the lower bound is equal to $O(\log_2 n)$ steps, where $O[f(n)]$, according to [3], denotes a set of all values $g(n)$ for which there exist the positive constants c and n_0 that the inequality $g(n) \geq c \cdot f(n)$ is fulfilled for all values of $n \geq n_0$. If the orgraph is represented by a matrix of contiguities, then the upper bound of the complexity of the algorithm has the estimate that at least $O(\log^2 n)$ steps must be fulfilled in the case of n^3 processors.

In cases when only a limited number of K parallel processors is available to the investigator, one can use several procedures. For example, each time when an arc leading to a new vertex is found in the algorithm of [8], the search continues from the "new" vertex and the procedure does not return to the "old" vertex until all the arcs of the "new" vertex have been considered. We shall compare this sequential procedure below to the possible versions of procedures based on parallel calculations.

Let us assume that the investigator has at his disposal K processors which must be used to find a set of all the nonrepeating orcycles for the graph presented in the form of a list. To analyze the procedures of sequential and parallel calculation, let us assume that each "visit" to the vertex is an active operation and that the procedures are compared only with respect to active operations. When a vertex is selected for a search, each processor checks one emerging arc and establishes whether a given arc is included in a new vertex. A partial list of all the new vertices found is then formulated. These partial lists are subsequently combined and added to the list of new vertices and the addition-operations are also regarded as active.

It is shown in [3] that if only active operations are taken into account, then the upper bound for any sequential search operation comprises $S_1 = \sum_{i=1}^{n} (\rho_i + 1)$

steps, where ρ_i is the local power of the emerging arcs of the i-vertex. It follows from these concepts that the optimum algorithm in which K parallel processors is used requires S_1/K steps.

Three different procedures [3] are used for unoriented graphs which can be used in somewhat modified form to find all the nonrepeating orcycles in the orgraph. The first of the procedures can be called a "K-deep search," bearing the following in mind: when the vertex is selected, K arcs emerging from this vertex are analyzed simultaneously, thus ensuring that all the processes are loaded. One of the just-found vertices is then selected and the search is continued from this vertex. A deep search is actually conducted but it is performed to the K-width at each step. Let us consider this procedure on the example presented in Figure 2.

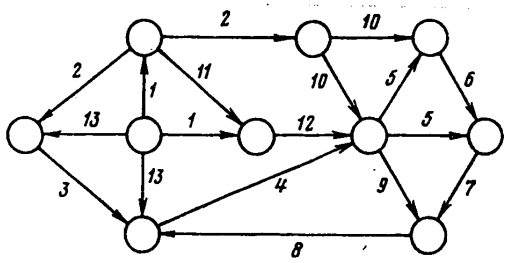


Figure 2. "K-Deep Search" Procedure at K = 2. The numbers denote the order of passage through the arcs

It is obvious that the number of steps in this procedure determines the total number of "visits" of all vertices i = 1, ..., n, where the number of "visits" to the vertex is calculated by the expression $(\rho_i + 1)/n$ and the number that indicates the number of steps required to connect new vertices is $(\log_2 K) + 1$.

Expressions are given in [3] to determine the number of steps in the sequential procedure and the "K-deep search" procedure and it is concluded that the advantages of the latter appear if the inequality $2m/n \geq (\log_2 K) + 17$ is fulfilled and increases with an increase of the ratio 2m/n.

In the second procedure, when selecting the vertex, one must consider all the arcs emerging from it and only then turn to a new vertex (the procedure can be conditionally called a "wide and deep search" procedure), as follows from Figure 3. This procedure is more effective than the sequential procedure when the inequality $2m/n \geq (\log_2 K) + 7$ is fulfilled.

Finally, the third procedure envisions the following: when the vertex is selected, all the processors are engaged in a search for the arcs emerging from it. In this case the vertex located at distance 1 from the initial vertex x_0 is always considered before the vertex located at distance 1 + 1 from x_0 (Figure 4). The procedure is more effective than the sequential procedure if the inequality $2m/n \geq$ $\geq (\log_2 K) + 5$ is valid.

Analysis of the procedures using the parallel calculations model to which rather rigid restrictions are applied, indicates an increase of the effectiveness of parallel procedures as the complexity of the problem increases, i.e., as the complexity of the structure of the orgraph being analyzed increases. Let us note in this regard the more complex problem of analyzing the connectedness of a so-called

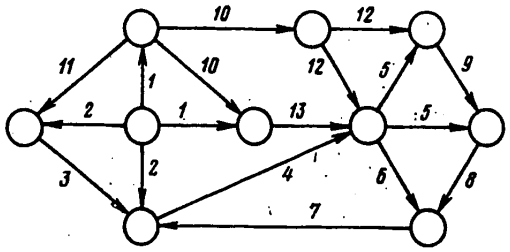


Figure 3. "Wide and Deep Search" Procedure at K = 2

diffuse graph [9], which is a complete weighted orgraph to whose arcs are given weights in the range [0, 1]. The use of parallel calculations in this problem is the only possibility of finding the result since sequential procedures essentially cannot be realized even with a comparatively small (8-10) number of vertices. The advantages of parallel procedures are more clearly marked here since, first, the orgraph is completely "filled," i.e., it has all possible arcs, and second, one must operate with the weights of arcs (compare and calculate the minimum and maximum weights of different paths and so on), which causes considerable additional expenditures of machine time.

One should point out problems which probably cannot be solved in principle by using the described procedures: graph numbering problems, i.e., operational construction and identification of all topologically different structures of orgraphs [10]. It seems probable that even the most indirect methods of parallel calculations and the use of a large number of parallel processors will help one to solve this probelm within a practically acceptable time for those complex systems which number several thousand components (vertices of an orgraph).

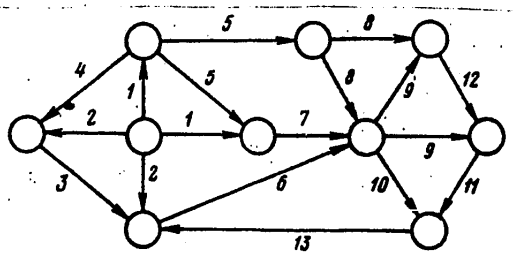


Figure 4. "Wide Search" Procedure at K = 2

Thus, there is an enormous class of important problems based on orgraphs which are ineffectively solved or generally not solved if one uses sequential procedures. These problems should primarily include problems of identification of orgraphs and investigation of the connectedness of both orgraphs (determination of the set of unrepeated paths and orcycles, finding connected components, determining the shortest paths between any pair of vertices and so on) and of nonorgraphs and also finding externally and internally stable sets of vertices, determination of the nucleus of a graph, calculation of radius and diameter, performing different operations on graphs and a number of others. Considerable success can be achieved in solving

all these problems only if one uses effective methods of parallel calculations, but to do this, investigators should have at their disposal not only the corresponding algorithms but suitable computer equipment as well. Of course, the increase in the speed and volume of the memory of computer systems and also the use of new configuration of the latter, for example, multiprocessor computer systems, may expand to a significant degree the class of problems based on orgraphs which can be solved with practically acceptable expenditures of resources, but it is more probable that the capabilities of computer systems constructed on the basis of microelectronics are still limited.

The fact is that parallel calculations in investigation of the structure of orgraphs generate a large number of comparatively short and noninvariant branches, the number and length of which vary as a function of the specific problem. Multiprocessor calculating systems for a load consisting of a large number of small parallel problems do not ensure the required increase of effectiveness and moreover, have a number of other disadvantages (complex systems programming, difficulties in program correction, the occurrence of conflicts of processors during simultaneous access to the same resource and so on). This circumstance is related to the fact that more significant advantages of parallel calculations can be achieved with parallel operation when an essentially unlimited processor resource and a very large memory capacity are acceptable in which the contents of several sections can be varied simultaneously, while the current results are stored but not entered in the memory after each step, as in the model of parallel calculations considered above. This does not actually mean that there must be the capability of seemingly simultaneous generation of the structure of parallel calculations as a whole or in other words, that the idea of the isomorphism of the structure of the calculating system and the facilities analyzed by using it should be realized.

Optoelectronic computer systems, we feel, more fully correspond to these requirements since optical processing of information promises fundamental solution of the problem of developing internal storage devices of far greater capacity and gigantic archive memories with rapid access and also input-output devices with large carrying capacity [11]. Moreover, the optoelectronic medium itself permits one to construct those computer systems which realize the idea of invariant contribution of the model of the facility being analyzed to the computer system [12]. These systems not only have high productivity but the logic (functional) capabilities in them also increase significantly due to parallelling of the basic calculating program to parallel branches. Matrix (parallel) synthesis of the required parallel structures of calculations of large dimensionality, operational restructuring of them and generation of a chain of signals (multiple echo) in response to a single input signal [12] are possible, which is very promising to investigate the structure of orgraphs since it permits one to reproduce all the information recorded previously. Thus, the procedures that search the vertices for the "load" of processors are simplified. Optoelectronic computer systems obviously permit one to realize more simply so-called "pattern logic" operations [13], i.e., operations with entire information files in parallel without sequential scanning. This means that the work to investigate complete graphs represented by a matrix of contiguities is considerably simplified.

The given concepts with respect to optoelectronic computer systems permit one to state that one can expect fundamental solution of timely problems of graph theory,

among which the problem of investigating the structure of orgraphs occupies a central position, only after development and introduction of these systems.

BIBLIOGRAPHY

1. Grundspen'kis, Ya. A., Z. P. Markovich and Ya. Ya. Osis, KIBERNETIKA I DIAGNOSTIKA, No 5, Riga, Zinatne, 1972.

2. Buslenko, N. P., V. V. Kalashnikov and I. N. Kovalenko, "Lektsii po teorii slozhnykh sistem" [Lectures on the Theory of Complex Systems], Moscow, Sovetskoye Radio, 1973.

3. Arjomandi, E. and D. G. Corneil, SIAM JOURNAL OF COMPUTING, Vol 7, No 2, 1978.

4. Harari, F., "Teoriya grafof" [Graph Theory], Moscow, Mir, 1973.

5. Grundspen'kis, Ya. A., KIBERNETIKA I DIAGNOSTIKA, No 5, Riga, Zinatne, 1972.

6. Acho, A., J. Hopcroft and G. Ullman, "Postroyeniye i analiz vychislitel'nykh algorithmov" [Construction and Analysis of Calculating Algorithms], Moscow, Mir, 1979.

7. Prabhaker, M. and N. Deo, SIAM JOURNAL OF COMPUTING, Vol 5, No 1, 1976.

8. Tarjan, R. E., SIAM JOURNAL OF COMPUTING, Vol 2, No 2, 1973.

9. Takeda, E., "Voprosy analiza i protsedury prinyatiya resheniy" [Problems of Analysis and Decision-Making Procedures], Moscow, Mir, 1976.

10. Arutyunov, P. A., "Topological Analysis of Chains Using Random Number Structures," MIKROELEKTRONIKA, Vol 4, No 6, 1976.

11. Svidzinskiy, K. K. and V. P. Lavrishchev, "Problems of Constructing Optoelectronic Information Processing Systems," MIKROELEKTRONIKA, Vol 2, No 1, 1973.

12. Bykhovskiy, V. K., PROBLEMY GOLOGRAFII, No 2, Moscow, MIREA, 1973.

13. Edel'shteyn, L. A., ZARUBEZHNAYA RADIOELEKTRONIKA, Vol 5, 1974.

6521
CSO: 1863/54

PROGRAMMING PL/1 COMPUTER SYSTEM

Moscow PROGRAMMIROVANIYE NA PL/1 OS YES in Russian 1979 (signed to press 29 Oct 79) pp 3-4, 266-269

[Foreword and table of contents from book "Programming in PL/1 OS YES", by M. I. Auguston, R. P. Balodis, Ya. M. Barzdin', E. A. Ikauniyeks and A. A. Kalnin'sh, Izdatel'stvo "Statistika", 60,000 copies, 271 pages]

[Text] The algorithmic language PL/1 (Programming Language/One) was produced during the 1963-1966 period by a committee of computer users and IBM representatives. PL/1 belongs to the generation of languages which followed ALGOL-60, FORTRAN and COBOL. It not only unites the possibilities of these languages, but it also gives the programmer a number of additional possibilities. The language version for the OS YES (PL/1 OS YES) operational system is fairly rich, and the compiler is sufficiently perfected and maintains an optimizing mode which produces operating programs slightly inferior in its qualities to the program composed in ASSEMBLER language.

Nevertheless, PL/1 language has one important shortcoming--it is very complex. Complete documentation for PL/1 OS YES fills over 2,000 pages. This significantly complicates the wide use of this language. At the same time, by no means are all the facilities of the language, as a rule, used for programming separate classes of tasks. One of the most important classes of tasks, where the PL/1 application can be the most effective, are the data processing tasks. A sub-mass of PL/1 OS YES, orientated in particular towards this class of tasks, is described in this book. We hope that it will be sufficient for practical programming. We included also in the book information about OS YES which is necessary for operations with PL/1 OS YES (for an expanded summary of OS YES see bibliographic references 2, 4 and 5).

One of the reasons for the complexity of expounding the PL/1 language is that there is no simple model upon which it would be possible to explain the semantic of the language. One such model, the so-called idealized computer ("PL/1 Computer"), is proposed in this book for a selected sub-mass of the language. In our view, this significantly facilitates expounding the material, maintaining mathematical accuracy.

This book was written as a practical manual and is intended for programmers becoming acquainted with PL/1 OS YES for the first time. We endeavored to explain the elements of the language and of the operational system encountered in it in such a way that it would be sufficient for practical programming. When composing programs, we recommend avoiding cases in the book which say "result uncertain" or "result not explained in more detail". A subject index and a list of PL/1 (supplement 6) facilities used are cited at the end of the book for convenience.

Operational documentation of the YES computer was used during the writing of the book. Vague constructions and examples were checked on the 4.0 OS YES version. From other manuals on PL/1 one can mention bibliographic references 1, 6, 7, 8, and 9.

The exposition plan and coordination of final editing of the book were carried out in the main by Ya. M. Barzdin' with the participation of A. A. Kalnin'sh. General input-output and organization of data collection were written in the main by R. P. Balodis. In other respects, all the authors bear equal responsibility for the contents of the book.

The authors are grateful to the reviewer of the book Professor V. N. Lebedev for the valuable criticisms, and also to K. M. Podnieks, Yu. V. Borzov, and other members of the Computer Center of Latvian State University imeni P. Stuchka, the advice of whom helped substantially to improve the account.

		Page
Foreword		3
Chapter I.	Introduction to PL/1	5
1.	Program Elements. Recording the Program on a Form	5
2.	The Language Alphabet	8
3.	Identifiers	8
4.	Idealized Model of a Computer	9
5.	Metadesignations	11
Chapter II.	Data	12
1.	Numbers and Lines. Constants	12
2.	Variables. Types of Variables	14
3.	Decimal Data With Fixed Point DECIMAL FIXED	15
4.	Binary Data With Fixed Point BINARY FIXED	16
5.	Decimal Data With Floating Point DECIMAL FLOAT	17
6.	Binary Data With Floating Point BINARY FLOAT	18
7.	Line of Symbols Type Data CHARACTER	19

8. Line of Bits Type Data BIT 19
9. Line of Variable Length Symbols Type Data CHARACTER VARYING 20
10. Line of Variable Length Bit Type Data BIT VARYING 20
11. Pattern PICTURE 21
12. Variable Controls of Program POINTER and LABEL 24
13. Masses 24
14. Structures and Masses of Structures 27
15. Methods of Abbreviating Operator Recording DECLARE 30
16. Attributes Assigned by Silence 31

Chapter III. Operations With Variables and Appropriation Operator 33

1. Comprehending Exceptional Situations 33
2. Simplest Case of Appropriation Operator 33
3. Operations. Arithmetic Operations 38
4. Comparison Operations 41
5. Logic Operations 42
6. Coupling Operations 43
7. Expressions 43
8. Common Case of Appropriation Operator for Problem Variables 47
9. Appropriation Operator for Aggregates of Variables 47
10. Attribute INITIAL 48

Chapter IV. Built-In Functions 51

1. Comprehending Built-In Functions 51
2. Arithmetic Built-In Functions 52
3. Mathematical Built-In Functions 54
4. Built-In Functions for Line Processing 55
5. Built-In Functions for Mass Processing 56
6. Special Built-In Functions 57
7. Pseudovariables 57

Chapter V. Operators of Program Control 59

1. Execution Order of Operators. Indices. Operator GO TO 59
2. Operator IF 60
3. Operator DO 62
4. Operators END and STOP 65

Chapter VI. Program Structure. Procedures. Memory Control 66

1. General View of Program 66
2. Procedures Call. Main Procedure 67
3. Program-Procedure Call 69
4. Function-Procedure Call 73

5. EXTERNAL Variables 76
6. Program Execution 76
7. Execution of Sub-Program-Procedure and Function-Procedure 77
8. Transmitting Parameters of Variable Length 79
9. Transmitting Parameters of Other Kinds 80
10. Conversion to Standard Programs 81
11. Internal Procedures 82
12. Program Examples 84
13. Base Variables and Memory Control 85

Chapter VII. Means of Exceptional Control 90

1. Comprehending Exceptional Control 90
2. ON Operator 91
3. Situations Associated With Computer Operations. Situation ERROR 94
4. Means of Exceptional Control for Program Deferral 96

Chapter VIII. Files 100

1. Basic Concepts 100
2. Types of Files 101
3. Work Modes With Files 107
4. File Declaration in a Program. Standard Files SYSIN and SYSPRINT 108
5. Opening and Closing a File 109

Chapter IX. Input-Output by Flow 112

1. General Information. Input-Output in the LIST Mode 112
2. Input-Output in the EDIT Mode 115
3. General Format of Operators of Input-Output by Flow 121
4. Operators GET STRING and PUT STRING 122

Chapter X. Input-Output by Recording. Exceptional Situations Associated With Input-Output 124

1. General Information on Input-Output by Recording 124
2. Input-Output of CONSECUTIVE Files 126
3. Sequential Data Input-Output from INDEXED Files 127
4. Direct Access by Recording for INDEXED Files 131
5. Working with REGIONAL (1) Files 133
6. Exceptional Situations Associated With Input-Output 134
7. Communication With the Computer Operator During Program Execution in PL/1 138

Chapter XI. OS YES Operational System and Task Control Language 140

1. Introduction to OS YES Operational System 140
2. Task Control Language. Cataloguing Procedures 142
3. Syntax of Operators of Task Control Language 147
4. Input and Output Flow 148
5. Peculiarities of Executing Tasks in the PCP, MFT and MVT Modes 149
6. JOB Operator 151
7. EXEC Operator 153

Chapter XII. Data Set-Ups and DD Operators 156

1. External Carriers of Information 156
2. Brief Review of DD Operator Parameters 158
3. Data Input From Punched Cards 163
4. Data Output on Punched Cards 165
5. Data Printing 166
6. Data Set-Ups on Magnetic Tape 167
7. Developing Data Set-Ups on Magnetic Tape. Addition of Data at the End of a Data Set-Up 170
8. Reading Data From a Data Set-Up on Magnetic Tape 172
9. Structure of Magnetic Disks 172
10. Developing a Sequential Data Set-Up on Magnetic Disk. Addition of Data at the End of a Data Set-Up 175
11. Developing a REGIONAL (1) Data Set-Up 177
12. Structure of Sequential-Index Data Set Ups 178
13. Developing a Sequential-Index Data Set-Up. Addition by Recording at the End of a Data Set-Up 180
14. Reading Data, Addition and Correction by Recording for All Kinds of Data Set-Ups on a Disk 182
15. Data Set-Ups on Punched Tapes 182
16. Data Input From Punched Tapes 183
17. Data Output on Punched Tape 184
18. Library Data Set-Ups 185
19. Developing Libraries and Adding Sections to the Library 186
20. Data Input From a Library Section 188
21. Additional Possibilities of the DD Operator 188
22. Brief Summary of Other Possibilities of the Task Control Language 197

Chapter XIII. PL/1 Compiler 198

1. Data Set-Ups Used by a Compiler 198
2. Transmitting Parameters to the Compiler 200
3. Parameters of the Compiling Mode 201
4. Compiling Listing 202
5. Pack Compiling 207

Chapter XIV. Communication Editor and Loader 209

1. General Information. Simple Use of the Communication Editor 209
2. Operating Schema of the Communication Editor 210
3. Transmitting Parameters to the Communication Editor 212
4. Data Set-Ups of the Communication Editor 213
5. Formation of Structures With Overlap 218
6. Examples of Using the Communication Editor During Operations With Libraries 222
7. The Loader 224

Chapter XV. Program Execution in PL/1 228

1. Program Execution by Means of Cataloguing Procedures 228
2. Execution of the Loading Module From the Library of Loading Modules 231
3. Optimization of Execution Time 233
4. Optimization of Memory Capacity Necessary for Program Execution 235

Supplements:

Supplement 1. Table of Codes DKOI and KPK-12 237

Supplement 2. Table of Codes DKOI and KOI-7 238

Supplement 3. Example of Program Execution on the Computer 239

Supplement 4. Utilized Key Words of PL/1 (Permissible Abbreviations are Indicated in Parentheses) 245

Supplement 5. Utilized Key Words of OS YES (Permissible Abbreviations are Indicated in Parentheses) 247

Supplement 6. Utilized Facilities of PL/1 250

Supplement 7. Brief Review of the Possibilities of Complete PL/1 Not Included in the Described Sub-Mass 258

Literature 261

Subject Index 262

9889
CSO: 1863/75

UDC 681.3.06

SOFTWARE OF TYPE M5000 COMPUTER COMPLEX AND DESCRIPTION OF ITS ELEMENTS

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 pp 1-3

[Article by V.N. Popkova, engineer]

[Text] Facilities for Automating Programming

The M5000 DOS [disk operating system] places at the disposal of a programmer a large set of facilities making it possible to automate the entire process of solving a problem on a computer.

The programs of the M5000 DOS operating system are organized into libraries (with the exception of the initial loading program and the supervisory program part) and are placed on a magnetic data medium. The library structure of the M5000 DOS makes it open-ended. The operating structure of the M5000 DOS offers the programmer programming languages oriented toward various classes of problems: the machine-oriented ASSEMBLER language, the procedure-oriented COBOL language and the problem-oriented RPG [report program generator] language. ASSEMBLER is the most flexible and universal facility for writing programs and its capabilities are expanded by macrolanguage facilities. In the ASSEMBLER language the possibility is provided of implementing auxiliary functions which help the programmer to check and document programs, to control the distribution of addresses, to allocate data and labels, and to generate and control translation of the program itself. Conditional compilation and the macrolanguage represent the ASSEMBLER language portion of the M5000 DOS.

The macrolanguage makes it possible for the programmer to input his own statement--a macroinstruction--which can be used in the program if it has first been represented in the form of a specific sequence of ASSEMBLER language statements. Thus, the macrolanguage simplifies the coding of programs, making possible the use of a standard sequence of statements.

Conditional compilation facilities make it possible for the programmer to write in ASSEMBLER language statements compiled as a function of compilation conditions computed at the moment and can be used both within macrodefinitions and outside of them, i.e., without using macrolanguage facilities.

The COBOL language is oriented toward descriptions of algorithms for processing business and economic information. Forming its basis are the recommendations of

the American Standards Association and the recommendations of the group for algorithmic languages for processing economic information of CEMA countries. The COBOL of the M5000 DOS operating system consists of a core and 10 modules (processing of tables; sequential, relative and indexed input/output; sorting-merging; report generator; segmentation; library; debugging; program interfaces). All modules are of the second level. The language utilizes more than 400 key or extra words.

The RPG language is designed for writing algorithms for solving problems relating to the creation and printout of various reporting forms, such as summaries, lists, records, etc. A distinctive feature of problems solved by using the RPG system is the fact that in the process of solving them the main share of machine time is due to the processing of large input files and the formation of corresponding output files. Here the calculation part occupies a relatively small place in the total number of problems and the solution results must usually be output in the form of printed documents. The problem-oriented language of the IBM/360 system is used as the basis of the RPG language in the M5000 DOS operating system.

It is permissible to write programs in all languages by using both the Russian and English variants of the language.

The presence of several programming languages and the step-by-step nature of the preparation of programs used in the M5000 DOS make it possible for the programmer to divide his task into parts--modules--and to select for each module the most appropriate programming language. The M5000 DOS operating system makes possible the independent translation of each source module. A program written in any of the programming languages indicated is translated into a relocatable module which is obtained in a format which is common to all translator routines of the system. The specific properties of the source language are lost after translation.

Relocatable modules are processed by an interface editor and they become available to run a program--absolute modules (phases). The unification of relocatable modules into phases takes place regardless of when and from which programming language a specific module has been translated. The interrelationship of phases of a program at the moment of running is determined for each task.

A program can either be of simple structure, when it is called in its entirety to the main memory to be run, or have overlapping, when for purposes of reducing the capacity of the main memory used by the program while it is run the program is divided into segments which overlap one another in the main memory when the program is run. Program segments can be independently written and translated into relocatable modules. In turn, available subroutines individually prepared in the form of relocatable modules can be used within segments.

Preparation of a Program for Running

In the M5000 DOS all programs, depending on which stage of preparation they are in (source module, relocatable module, phase), can be stored in a corresponding module library: source libraries (IB's) with program modules written in the source programming languages; relocatable libraries (PB's) with relocatable modules in the interface editor program language, representing the result of the translation of

source modules; and absolute libraries (AB's) with program phases in machine language available to run. A disk storage serves as the storage for all libraries. The sizes of M5000 DOS libraries are selected by the manufacturer as a function of the number and size of elements which are to be put into the library.

The presence of an absolute library is obligatory for the functioning of the M5000 DOS operating system, since in it in the form of phases are contained such components of the DOS as nonresident supervisor and executive routines, translators, service routines, and also user programs available to run. Any program is loaded into the main memory to be run only from the absolute library.

Each library is in the form of a library file consisting of two parts--the body and the heading of the library--and is designed for storing elements of the library (body) and information on these elements. A set of programs under the general name of the librarian, which come under the heading of service programs and perform the functions of correction and copying and service functions, has been designed for working with libraries.

Under the heading of service programs also comes the interface editor, designed for processing the results of translation for purposes of obtaining programs available to run. Its functions are as follows: the assembly of a program from a number of relocatable modules; adjustment of the text of an assembled program for working in a specific region of the main memory; and placement of an edited program into the absolute library. In the formation of each phase the editor does the following:

Puts together the text of a phase from relocatable modules and individual program sections of relocatable modules, whereby the set of these and their relative position are indicated by the programmer.

Automatically includes in the text of a phase modules from the PB, if the need for this arises.

Adjusts the text of a phase according to the loading address specified by the programmer.

Introduces changes into the text of a relocatable module through correcting cards.

Forms links between modules when they are joined into a phase.

Forms the address for input into each phase.

The phase formed is placed by the interface editor program into the body of the absolute library and information on it into the absolute library's heading. The program obtained as the result of editing has an absolute format and in the simplest case consists of a single phase. Programs of large size usually include several phases which can replace one another in the main memory.

Control Program

The following are the characteristic traits of operating systems: enabling the continuous performance of work; preparation of tasks for execution; reacting to

software and hardware error situations; enabling the simultaneous operation of various input/output units and of the processor together with them; and the presence of a set of facilities for the automation of programming and of program debugging facilities. These same traits are characteristic of the M5000 DOS, in which the greater portion of these functions is implemented by control programs, i.e., the initial loading program, the supervisor, executive and the input/output control system.

Before starting to work with the computer, the computer itself and the operating system must be brought into a state of readiness. The initial loading program performs the initial preparation of the system for operation. It clears the main memory, receives information on the position of the system, performs several auxiliary functions and then enters the core of the supervisory program into the main memory.

The supervisory program is the central control program, in the form of a set of routines put into action by means of interrupts originating on the request of service routines or as the result of a situation formed in the process of the complex's operation. It controls the entire computing process in the computer and implements the following functions: the processing of software and processor errors; the performance of input/output operations at the request of a program; organization of the simultaneous operation of the processor and an input/output unit; processing of input/output unit errors; organization of communication with the operator; and the performance of actions by request of the program (a supervisory program macroinstruction). The supervisory section--the core--implementing the functions most often used is permanently in the main memory. The other section of the supervisory program is organized in the form of transit phases stored in the absolute library and called to the main memory as needed.

The M5000 DOS executive routine prepares the system for performing each individual task and organizes the package processing of tasks. It accomplishes adjustment of the system for the needs of a specific program, loading of this program from the abolute library into the main memory, and the transfer of control to the program loaded. The executive routine receives the instruction for the performance of these operations by means of control instructions. The executive routine constantly reads and executes these instructions once it is started. This organization of the work of the M5000 DOS executive routine makes it possible to accomplish the package processing mode.

For the purpose of performing input/output operations it is necessary not only to program direct accessing of peripheral devices itself, but also to provide for cases of error and unusual situations even in cases when their origin is of low probability. In the M5000 DOS there is a strict distribution of duties between the control program attending to input/output (the supervisory program) and the programmer. The control program (the core of the supervisory program) takes upon itself the implementation of such input/output functions as determination of the state and the assignment of instructions to units, checking the quality of input/output operations performed, organization of the simultaneous operation of input/output units, the starting of input/output operations, and the processing of input/output interrupts, including interrupts caused by the errors of units. And

the programmer has the duty of preparing the program for the required input/output operation (placing file entries on external media, describing input/output operations) and of informing the control program of the need to run it.

This method of input/output programming makes possible flexibility in working with peripheral devices but has the disadvantage that it forces the programmer to know in detail the operation of these devices. Therefore, the M5000 DOS in addition offers the programmer a higher level of input/output programming--a logic system for controlling input/output, in the form of a set of standard procedures for realizing all input/output functions, which makes it possible for the programmer not to go into the details of input/output, but to concentrate his attention entirely on the logic functions of the system itself.

The logic system for controlling input/output processes files of data, performing the unification of entries into a block and their separation from a block; readdressing of the input/output region if the file uses two input/output regions; organization of the simultaneous operation of input/output units with the operation of the processor; processing of the "End of File" state; and the checking and recording of information on files. The logical content of the data, their format and the organization of data into files are important for it.

There are three methods of accessing in the M5000 DOS: sequential, direct and indexed sequential. Sequential access is applicable to files in all units (magnetic tape and disk, punched tape, punched cards, printout) and indexed sequential and direct access only to disk files.

Standard Facilities

The M5000 DOS operating system has a number of special facilities and programs which facilitate the work both of the programmer and operator:

PODPA, designed for preparing magnetic disks and tapes for work, checking the state of these media and labeling them.

KOPMD, for creating copies of disk packs and checking the agreement of copies with the original.

PEChMDL, for printing out the contents of a region of a magnetic disk and the zone of a magnetic tape.

ZAMMD, for replacing the contents of one sector of a magnetic disk.

SRML, for comparing two files located on different tape reels and for establishing their identity.

The general-purpose RYePS, for copying information on the physical level from some media onto others.

PPASP, for printing out the labels of files located in packs in one or more disk storages.

KARL, for copying a file from punched cards onto magnetic tape.

PEChPK, for outputing a punched card file to an alphanumeric printer.

KORP1, for copying the contents of magnetic disk packs onto magnetic tape and vice-versa.

In the M5000 DOS there are a number of facilities provided for ordering files of records, i.e., for arranging them in relation to the values of some of the quantities comprising them, so-called control fields, viz., a sorting-merging program generator. It consists of two groups of programs: 1) a set of modules in the relocatable library from which by means of the interface editor program it is possible to construct independent sorting or merging programs of various structures; and 2) sorting and merging subroutines in the absolute library, which are accessed via sorting and merging macroinstructions.

For the purpose of forming an operating system with the properties most suited to both the configuration of the complex and the requirements placed on it by the user, system generation facilities are employed which make it possible to replan the size and distribution of system files in the complex's external storage, as well as the size and contents of libraries, and to plan a supervisory program with assigned functions by assigning to the statements of the supervisory program's generation macroinstructions values which differ from the standard.

Copying programs available in the M5000 DOS make it possible to copy sequentially organized files from some data media onto others. By running copying programs it is possible to copy, rearrange, reblock, supplement (only for disks) and display (printout only).

Thus, the M5000 DOS, representing a combination of control programs (supervisory program and executive, initial loading program and input/output control system) and service programs (translators, interface editor, librarian, sorting-merging program generator, DOS generation system and copy programs and special programs), makes it possible to automate the process of solving a problem on a computer and to orient the system toward solving specific problems and satisfies the needs of a wide range of users.

8831
CSO: 1863/80

UDC 681.325.5

SYSTEM APPROACH TO AUTOMATED DESIGN OF MICROPROCESSOR COMPUTERS

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 pp 3-4

[Article by V.A. Vishnyakov, candidate of technical sciences]

[Excerpts] The creation of problem programs for realizing the goals stated, i.e., cross software, makes it possible to reduce considerably within a given period the time for developing and introducting microprocessor systems.

Relatively recently the development of design system software was based on the creation of packages of applied programs; however, the experience of developing these systems in the USSR and abroad demonstrated that this form for designing software does not provide the required universality. The difficulty of working with individual applied programs, the lack of dialogue facilities for debugging the software of microprocessor systems, the complexity of interacting with the data base in creating microprocessor systems, questions of reliability, and the organization of multitask operation in the design process require a comprehensive approach to designing microprocessor systems, i.e., the creation of an integrated SAPR [automated design system]. Such an SAPR should satisfy the requirements of universality and adaptability and have control over the design process, a developed data base, facilities for organizing multiprogram operation, and automatic and dialogue modes.

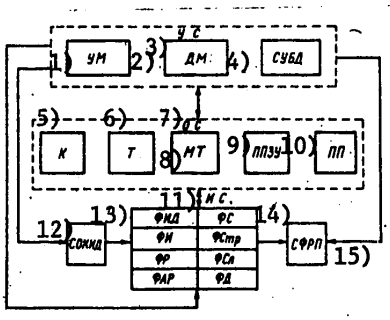


Figure 1.
[Key on following page]

Key:

1. Control executive routine
2. Dialogue executive routine
3. Control system
4. Data base control system
5. Compiler
6. Translator
7. Processing system
8. Microtranslator
9. ROM programmer
10. Applied programs
11. Information system
12. Source data description and coding system
13. Source data file; information file; working file; archive file
14. Reference file; structure file; service file; dialogue file
15. Design results formation system

The structure of an SAPR for microprocessor computers has been developed by taking these requirements into account (fig 1).

This system solves problems in designing hardware and software for computers of average capacity. Further designing, including the design and testing of prototypes, preparation of the coding system, checking the logic of the software and testing the equipment, is performed on a microcomputer automated design unit. The control and information sections of the SAPR have been developed at the present time and there are also problem routines for automating several stages in the development of microprocessor systems. System modules have been implemented in the PL/1 and ASSEMBLER languages.

8831
CSO: 1863/80

UDC 681.326

AUTOMATION OF DEVELOPMENT OF PROGRAMS FOR DATA RECORDERS

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 12, Dec 81 pp 4-5

[Article by A.M. Longin, T.A. Masal'skaya, T.A. Yepikhina and M.I. Abezgauz, engineers]

[Text] Data recorders of the RI-7501 and RI-7502 types, representing programmable self-contained units designed for gathering, arithmetic processing and recording of information, are widespread in ASU's [automated control systems]. The functions of recorders are as follows: the input of information from digital and alphanumeric keyboards, a data card, punch badge and conditionally constant control character unit; the output of information for printouts (documents), onto punched tape and into communication channels; arithmetic processing of input information (addition, subtraction and multiplication operations); the storage of data in registers of the recorder's memory for the duration of the running of a program; and the reception of information from a communication channel and the recording of it on a printer (for the RI-7501). Information recorders make it possible to control input information in terms of format, by the modulo-10 double input method. The memory of data recorders consists of three registers with a capacity of 128 bytes each.

The operating program for RI-7501 and RI-7502 data recorders is the sequence of operations which the recorder must perform for the purpose of forming a document, forming a message on punched tape and transferring it to the communication channel, and also for receiving a message from the communication channel. The programming system of data recorders represents a set of eight instructions, two of which have modifications. The sequence of program instructions is entered on a program punched tape (or on two punched cards if the length of the program exceeds 80 bytes) in a special code.

Certain difficulties in the process of "manual" programming of the work of data recorders originate when taking into account the features of the operation of the input and data processing unit and the 2-register memory, as well as in writing programs which economically utilize a small amount (128 bytes) of the storage register designed for storing the program. When an RI-7501 works with a computer through a communication channel, an RI-8901 group data recorder is used as the switch, imposing additional restrictions on the structure of the message which can be relayed through the communication channel. From what has been said it can be concluded that it is advisable to develop a method of automated programming for

the purpose of making possible the machine processing of primary documents of various forms.

A method suggested by the authors, called the "Program Generator for Data Recorders" (GPRI), which unites in itself the "manual" preparation of formalized source data with the automatic (in the computer) generation of an operating program for the data recorder, is described in this article. A diagram of the technological process of automated programming for a data recorder is shown in fig 1. The text of the program to be read by a human being is recorded in the output of the alphanumeric printer, 10, and codes to be read by the machine in the output of the punched card output unit, 9. The job of the designer, 4, is as follows: a) to create a form, 5, for formalized source data based on the printed document's form, 2, and a message format, 1, taking into account the instructions, 3; and b) to analyze errors discovered in the program generated and to correct the formalized source data or to order the correction of punching errors (can be lacking).

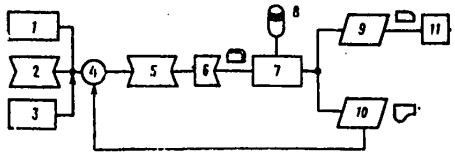


Figure 1. Diagram of Technological Process of Automated Programming for Data Recorders: 6--unit for preparation of data on punched cards; 7--computer (processor); 8--disk storage; 11--data recorder; (the remaining designations are given in the text)

The form with the formalized source data is a printed form containing three tables containing indicators identifying the program, as well as information on the operating modes of the data recorder and the kinds of check employed, and the characteristics of the particulars of the message to be formed. The form with the source data is at the same time the form for punching.

The GPRI per se is a program for any YeS [Unified Series] computer having a PL/1 compiler. The GPRI program consists of 20 phases, which makes it possible to run it on a computer with a RAM capacity of a total of 64K bytes. In the course of generation of the GPRI program for a data recorder the source data are reviewed repeatedly. The program generated is also checked several times for the object of reducing its size by eliminating redundant instructions formed at the first stage of generation. The generator of programs for data recording realizes all the capabilities of the RI-7501 and RI-7502 programming system.

The GPRI program makes possible the performance on a computer of a logical check of source data for purposes of revealing the most widespread errors (more than 20 types) committed both in filling in source data forms and in punching data on cards. When an error is detected, a hardcopy log containing the following information is printed on the alphanumeric printer: description and place of error, report of the violation of apriori known conditions and recommendations on the method of correcting the error. In the GPRI is provided the function of correcting errors of certain types, a report on this is output to the alphanumeric printer and the

program generation process then continues. If the designer does not agree with the corrections introduced by the GPRI, then he can correct the source data.

The result of the GPRI's work is the data recorder operation program, output in the form of a table on the alphanumeric printer, 10, and one (two) program punched card which is output to the computer punched card output unit, 9.

In addition to the generation program the GPRI includes a standard disk-to-card copy program. The necessity of using it involves the fact that this program makes it possible to output data onto punched cards not only in DKOI [EBCDI] code, but also in any other code. At the same time programs for RI-7501 and RI-7502 data recorders are entered onto punched cards in a special code. The GPRI discussed records the program for the recorder in the required form in the working file in the disk storage, and program punched cards are output by means of the copy program in keeping with the information recorded in the disk storage.

The GPRI is now at the experimental utilization stage at TsNIITU [Central Scientific Research and Planning and Technology Institute of Organization and Control Equipment] (Minsk). Data recorder operating programs for more than 10 ASUP [automated system for controlling an enterprise] projects have been created by its means. Inputs of time in programming the work of type RI-7501 and RI-7502 data recorders, determined on the basis of analyzing the traditional "manual" method of programming and the experience of using the GPRI, are presented below.

Table 1.

"Manual" Programming	
Thorough study of programming and features of the work of data recorders, in months	1
Writing and debugging a single program of average complexity, in hours	16
Programming by Means of the GPRI	
Study of fundamentals of programming and instructions for filling in a source data form, in months	1
Producing a program of average complexity, in hours	1.27
Including:	
Filling in the formalized source data form, in hours	0.5
Punching source data, taking into account control by the verification method, in hours	0.25
Generation of a program on the YeS-1022 computer, in hours	0.02
Analysis of program produced, correction of source data and repeated generation of the program when necessary, in hours	0.5

Since in creation of the GPRI all the aspects of programming recorders are provided for in it, including the source data check function, the programs generated do not contain logic errors, inevitable in "manual" programming. The program punched cards obtained from the computer do not have punching errors (within the limits of the reliability of the operation of the computer card output unit).

At the present time the GPRI is being interfaced with a program debugging terminal system developed at TsNIITU. With this the designer has gained an opportunity, omitting the step of punching source data, of inputing source data directly from a video terminal. Operation of the GPRI by means of a program debugging terminal system makes it possible to shorten the development cycle for programs for data registers because of elimination of the task interrupt time in the "waiting" phase of the job waiting line in the computer, which is inevitable when working in the package mode.

8831
CSO: 1863/80

APPLICATIONS

UDC 681.325.5:621.9-52

APPLICATION OF MODERN MICROPROCESSOR TECHNOLOGY IN PROBLEM-ORIENTED SYSTEMS

Kiev ELEKTRONNOYE MODELIROVANIYE in Russian No 4, Jul-Aug 81 (manuscript received 9 Dec 80) pp 45-52

[Article by V. B. Smolov, D. V. Puzankov and G. A. Petrov]

[Text] Designation and main characteristics of microprocessor equipment. During the past decade, the capabilities of using computer technology (VT) in all fields of science and technology was expanded considerably with regard to the intensive development and great achievements in the field of microelectronic technology, development of new principles of organizing information processing systems and development of software. The developers of different systems can use a gamut of computer equipment, beginning with powerful general-purpose computers and minicomputers to microprocessor sets and microcomputers.

The appearance of microprocessors (MP) as new means of computer technology that are a class of large integrated circuits (BIS) was determined on the one hand by the advances of modern microelectronics, which makes it possible to locate up to 10^6 components of a single crystal, and on the other hand by a desire to intelligently combine the advantages of universal and specialized computer equipment. It is known that specialized computer equipment can provide better operational and engineering characteristics (high speed, reliability, cost, mass-size parameters, consumed power and so on) than universal equipment when solving many problems of mathematical processing of information. However, development of specialized BIS is in most cases practically unfeasible due to the long periods and high cost of developing them when converting to solid-state equipment. The development of microprocessor complexes (MPK) as a new component base of computer equipment eliminated to a significant degree the contradiction between universal and specialized information processing devices.

The microprocessor is a functionally complete universal program-controlled device for logic and arithmetic processing of digital information. A class of microprocessor computer equipment, to which microprocessor complexes of BIS, microcomputers of various designation and multimicroprocessor computer systems are related, have been developed during the past decade on the basis of microprocessor BIS.

The high reliability, universality, small size and other advantages of microprocessor devices provide the capability of building them into objects, bringing the computer equipment as close as possible to an information sensor or receiver.

Table 1. Characteristics of Soviet Microprocessor Complexes

Number of Item	Name of Series	Number of BIS in Series	Technology	Notation of TsPE	Digit Capacity
1	K536	14	p-MDP	K536IK1	8
2	K580	8	n-MDP	K580IK80	8
3	K581	5	n-MDP	K581IK1	8;16
4	K583	12	I^2L	K583IK1	8
5	K584	8	I^2L	K584IK1	4
6	K586	9	n-MDP	K586IK1	16
7	K587	7	KMDP	K587IK2	4
8	K588	3	KMDP	K588IK2	16
9	K589	9	TTLSh	K589IK02	2

Number of Item	Cycle Time, μs	Consumed Power, mW	Operating Temperature Range, °C	Designation
1	10.0	1,000	10-55	S5 microcomputer
2	2.0	750	10-70	SM EVM, microcomputer
3	0.4	900	10-70	Elektronika-60 microcomputer
4	1.0	300	10-70	YeS EVM, microcomputer
5	2.0	300	10-70	Minicomputer
6	0.5	1,000	10-70	S5 microcomputer
7	2.0	50	18-70	NTs microcomputer
8	2.0	25	60-85	NTs microcomputer
9	0.1	900	10-70	YeS EVM, SM EVM

However, MPK can be used effectively to develop not only microcomputers but large computers as well.

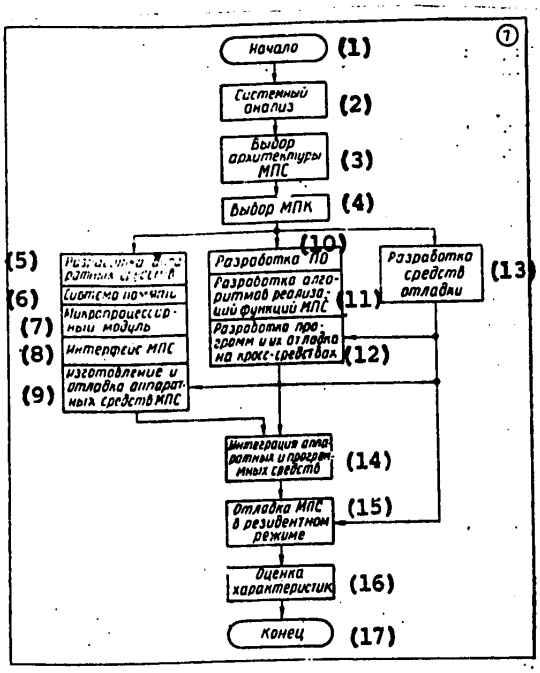
The characteristics of the main Soviet MPK, oriented toward various applications, are given in Table 1. As follows from the table, some MPK are the base for construction of microcomputers produced in our country. However, in many cases their characteristics do not meet the requirements of specific applications; therefore, it becomes necessary to develop microcomputers based on MPK that are adapted to the maximum to the suggested applications with minimum required set of hardware without any redundancy in calculating capacity, overall dimensions, mass and other characteristics. The main effect of MPK on design of problem-oriented information, computer and control microprocessor systems (MPS) also consists in this.

Characteristic features of designing microprocessor systems. Not only the component base but the approach to design of computer equipment changed with the appearance of microprocessors [1]. Analysis of the experience of developing systems based on MPK made it possible to determine the following main characteristics of MPS design.

1. Microprocessors are algorithmically universal modules whose functional orientation is determined by development of the appropriate software. In this case the interface of the given modules is organized from adequately developed recommendations and the periods for developing the hardware part of the system are considerably reduced. The use of microprocessors is a jump from small- and medium-integration circuits with a "rigid" structure in the field of programming to structural circuits of the processor-memory-commutator level.

2. The functional orientation of the system requires large expenditures of time and funds to develop applied and systems programs for each specific field. This is aggravated by the fact that microprocessor systems have begun to be used widely in different fields where computer equipment was previously not used due to the impossibility of using it or due to economic infeasibility. The practice of developing MPS shows that the fraction of economic expenditures on development of software is increasing continuously with respect to expenditures on development of hardware. For example, the cost of developing software comprises an average of 60 to 80 percent of the total cost of MPS developments.

3. The trend directed toward realization of software functions by hardware appeared. An example of this is development of BIS oriented toward control of peripheral devices such as disks, displays, printers and interrupt processing BIS and also development of BIS for mathematical calculations having a special set of instructions. The given BIS permit one to expand the field of MPS application and to increase their productivity. Thus, the INTEL Company has developed the microprocessor module 8087 oriented toward rapid arithmetic operations with floating points up to 64 digits and calculation of trigonometric functions. The distinguishing feature of auxiliary BIS is their programmability and also the fact that their complexity corresponds to or is higher than that of microprocessors. For example, the MP-8080 contains 5,000 transistors, the 8257 direct memory access controller contains 4,000 transistors while the BIS 8275 for connecting displays with raster scanning to the MP-8080 contains 15,000 transistors. This contributed to the development of systems design with expanders.



Key:

1. Beginning
2. Systems analysis
3. Selection of MPS configuration
4. Selection of MPK
5. Development of hardware
6. Memory system
7. Microprocessor module
8. MPS interface
9. Manufacture and debugging of MPS hardware
10. Development of software
11. Development of algorithms for realization of MPS functions
12. Development of programs and debugging of them on cross-devices
13. Development of debugging devices
14. Integration of hardware and software
15. Debugging of MPS in resident mode
16. Analysis of characteristics
17. End

specific difficulties since the microprogram developer must select the microinstructions format, thoroughly study the mechanisms of formulating subsequent microinstructions addresses, code transmission channels and possible integration during transmissions and must take into account delays and moreover there are frequently no microprogram writing, editing and debugging devices.

4. The use of microprocessors in systems design changes the process of debugging them with transfer of the center of gravity to debugging the software devices. This requires the presence of such basic software as a translator from Assembler language or a high-level language and editing, modelling and load programs.

5. The criteria of systems planning changed with the appearance of microprocessor sets of modules, the most important of which are:

the modular criterion that includes organization of MPS with minimum possible number of types of different components;

the contact structure between modules should be relatively regular while the number of contacts should be minimum, which reduces the requirements on the number of leads of the BIS module;

requirements of systems "support," testing and diagnosis of which becomes ever more difficult and important.

Designing the hardware and software of microprocessor systems. The main phases of designing microprocessor systems are presented in Figure 1. One of the most important phases is systems analysis, the purpose of which is to determine the requirements on the system being designed: to determine the set of functions which should be performed by the system, the required productivity, the cost boundaries in the development, manufacture and servicing phases of the system, and also to refine the critical functions according to the criteria of the system being developed, the analyze the composition of peripheral devices and their operating modes, to establish the nature of the preceding relationships of individual functions (subproblems) of the system, for example, series, parallel-series and intermittent contact and to determine the structure and "origin" of data and also the requirements on output data and the need to store it.

The requirements on the configuration of the system are determined and the problem of realizing the allocated functions by the software or hardware method are first resolved during the second stage according to analysis. The difficulty of solving the given problem frequently consists in the fact that there is a set of new applications of microprocessors which require investigations prior to decision-making.

Moreover, it frequently becomes encessary to investigate problems of module synthesis to implement functions by the hardware method due to the absence of the corresponding BIS in microprocessor complexes to realize the required functions.

The next phase of design is selection of the MPK, which must be accomplished on the basis of three main aspects.

1. Such microprocessor characteristics as the set of instructions and methods of addressing, digit capacity, number of general-purpose registers used, the presence and type of stack memory, the capability of processing interruptions and so on must be analyzed from the viewpoint of software development. The set of instructions must be selected and microprograms for realizing them must be developed if required for sectional microprocessors with microprogram level, which is related

2. On the basis of systems design, the completeness of the complex, which should contain, besides the microprocessor, PZU [External storage] and OZU [Internal storage], BIS modules for organization of the interface with peripheral devices, controllers for interrupt processing, control of direct access to the memory, bus shapers, buffer registers, on-off generator and systems controller, must be analyzed.

3. Such programs as a translator from symbolic language to a binary (entity) code, editing program and a modelling program without which the time and cost of MPS development increase considerably, is required from the viewpoint of software development and debugging. The given programs are usually developed for other computers and are called cross-devices for software development of the MPS being designed. Moreover, one must have a monitor, which permits one to control starting and stopping of the MPS from a terminal, to load the memory, for example, from papertape, to print the contents of the memory and registers of the microprocessors, to edit the contents of the memory cells and registers and to perform certain other functions, to debug programs in the so-called resident mode in the computer being designed. In this case part of the monitor functions can be used not only in debugging but in operating the system for operational "interference" of man, for example, in the control process: changing the parameters of the process, display of some variables or in other words for interactive operation of man and the control system.

The need to include the resident debugging phase is determined by the following factors:

the difficulty of modelling time-dependent and asynchronous events on cross-devices;

the speed of peripheral devices in a general-purpose computer is fixed and known and the intensities of the input flows are frequently unknown in systems for real-time applications, the moments of their arrival are determined by the controlled process and so on;

there is usually complex interaction of programs in real-time systems when using multiprogramming modes to process parallel processes and interruptions;

the increased requirement on systems reliability does not permit error correction during functioning.

Let us consider in detail the phase of software development of MPS, which is carried out in parallel to the phase of development and debugging of MPS hardware. The software and hardware are integrated and joint debugging of the MPS in the resident mode is accomplished after their completion.

Development of function realization algorithms. After determination of the functions which must be fulfilled in the MPS being designed, the individual function realization algorithms are developed. This requires organization of an interface to exchange information between individual software modules. The process of functional decomposition has several iterations existing up to the moment until the functions are written in terms of the algorithm, i.e., in the form a sequence of

determined and resultant effects, while information (input and output) for individual procedures is represented by a specific structure of data used in allocated function realization procedures.

Software module development. The given phase includes encoding specific functions in some language selected from the accessible languages: Assembler or a high-level language, the use of which has advantages and disadvantages. Unlike a high-level language, the number of lines of the initial program in Assembler is usually considerably greater--a single high-level language operator of the initial program generates an average of 5-10 machine instructions. Consequently, less time is required to write programs in this language and the programs are better documented and less complicated, but the volume of programs usually exceeds that in the number of instructions written in Assembler and this means that a larger memory and longer time are required to realize them.

Program translation. The translator translates the program into an entity code. In this case a syntactical and semantic check of the input program is made and diagnostic messages and listings of input and entity programs are printed out. The translator is a complex program. For example, the cross-assembler program of the series K-580 microprocessor language on the YeS-1033, developed at LETI [Leningrad Electrotechnical Institute] imeni V. I. Ul'yanov (Lenin) comprises approximately 1,700 operators of PL/1 language, while the required computer storage capacity to store the cross-assembler comprises 70 Kbytes.

Program modelling. The program is designed to debug entity programs of the microprocessor system. It coincides in its designation to the resident debugger, but may sometimes offer the user greater capabilities, for example:

output to the terminal of the complete route of executed instructions;

printing the time required to realize individual sections or the entire program;

stopping the program at checkpoints to retrieve diagnostic information;

printout of the contents of the microprocessor and memory registers, words of state of the program and so on.

Thus the program developed for the series K-580 microprocessor simulates one on-off operating cycle of the microprocessor system during 10 ms on the YeS-1033, the required storage capacity comprises approximately 77 Kbytes and the volume of the modelling program comprises on the order of 800 operators of ALGOL language.

The entity code must be retrieved for subsequent loading and debugging in the resident mode in the MPS after modelling of the entity program has been completed.

Use of microprocessor devices to solve some control and modelling problems. Analysis of control and modelling problems shows that the typical functions required for solving them are the following: data gathering from analog and digital signal sensors, conversion of analog signals to digital signals and vice versa, information

processing usually in real time, signal generation by the required law, reproduction of special functions, display and printout of results and so on.

Table 2. Computer Parameters

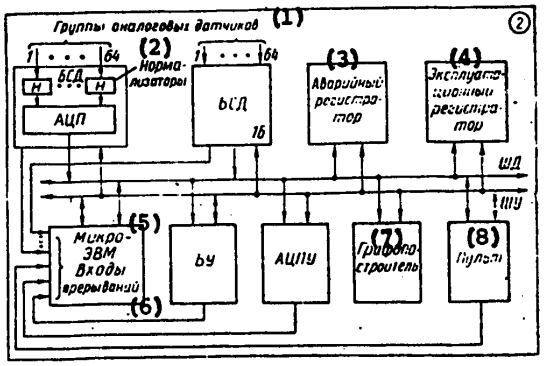
Name of Parameter	Value
Length of words to be processed	16 digits
Volume of addressable field of memory	64 K X 16
Number of addressable input-output devices	256
Number of instructions	90
Volume of stack	64 K words
Number of storage registers	2
Number of index registers	3
Number of memory addressing modes	12
Size of memory pages	256 words
Time required to fulfill instructions (with cycle of 2 s):	
addition with memory	2-3 μs
multiplication (mean value)	28 μs
division	30 μs
transfer	2-3 μs
Design version of processor	155 X 225 mm card
Power consumed by processor (at maximum clock frequency	8 W
Number of integrated circuits in processor	26
Mean cycles between failure for processor (without regard to power supply sources)	Approximately 16,000 hr

Investigations conducted at LETI imeni V. I. Ul'yanov (Lenin) since 1972 showed that the enumerated functions are typical for many information, control and computer systems, specifically production process, monitoring and measuring, scientific experiment automation and communications control systems [2-8]. Approaches to realizing the indicated typical functions are also considered briefly on examples of the development.

Measuring information gathering and processing. A block diagram of the system is presented in Figure 2. The basis of the system is a 16-digit microcomputer based on a series K-584 MPK [3]. Input information is entered into the microcomputer from 1,024 analog sensors and it is first normalized and converted to digital form by the analog-digital converter (ATsP) of the data gathering blocks (BSD). The results of processing are fed to recorders, a graph plotter and alphanumeric printer (ATsPU). The blocks are synchronized in the system by means of a control block (BU). The parameters of the microcomputer are presented in Table 2.

The software of the microcomputer consists of service programs used in development and debugging and applied programs. Symbolic coding language, cross-assembler, cross-interpreter and loader are related to the service software. The cross-devices are oriented toward the use of the YeS EVM [Unified computer system].

LY



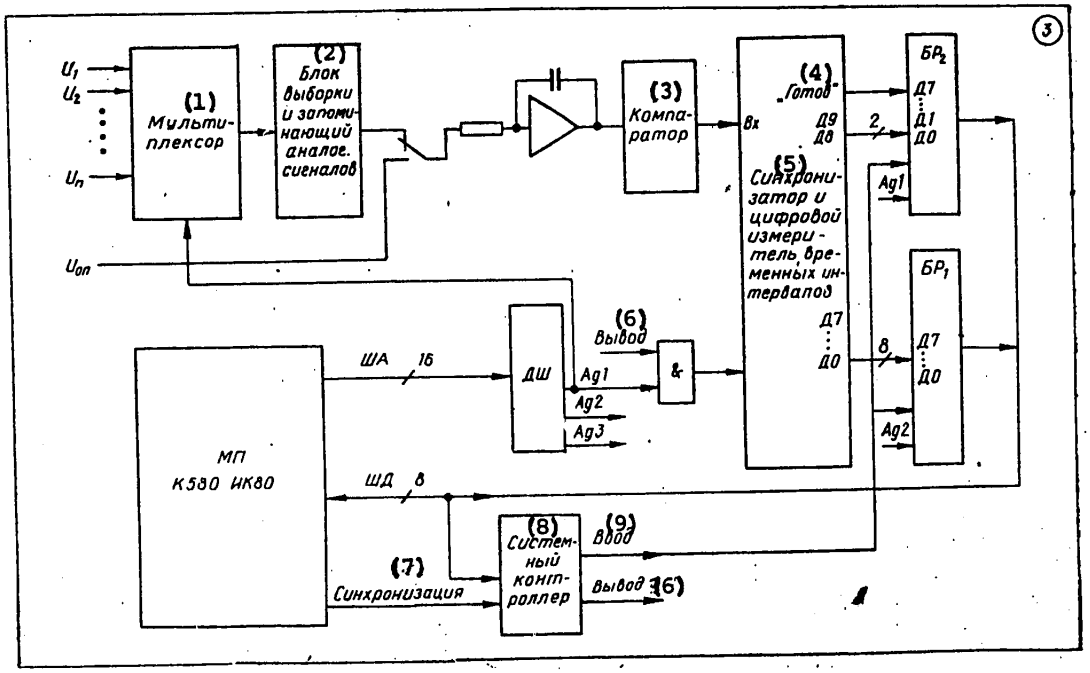
Key:

1. Groups of analog sensors
2. Normalizers
3. Emergency recorder
4. Operational recorder
5. Microcomputer
6. Interrupt inputs
7. Graph plotter
8. Console

Analog-digital information conversion. A block diagram of integrating an 10-digit ATsP based on alternate (dual) integration of the transformed and reference voltage from the IK-80 K-580 microprocessor is presented in Figure 3 [5]. The circuit operates in the "standby loop" mode, which envisions cyclic interrogation of ATsP readiness, which plays the role of a peripheral device, and code transmission. Two multimode buffer registers BR_1 and BR_2 with three-stable IR-12 K-589 outputs are used as the interface modules. The registers are selected by the microprocessor according to addresses Ad_2 and Ad_3 transmitted from the microprocessor through the address bus (ShA) at the moment when the data buses (ShD) of the microprocessor are tuned to receive information from the peripheral devices. The ATsP is triggered and information is collected after it is ready by means of a signal fed to the "Start" input of the ATsP and by analyzing the values of the "Readiness" signal from the output.

Solution of a system of differential equations. The structure of a parallel processor (PP) designed to solve systems of ordinary nonlinear differential equations having complex right side with large number of arbitrary functions of a single and greater number of variables is presented in Figure 4. The processor is oriented to operate in real time within the complex constructed on the basis of the SM-4 computer.

The parallel processor is constructed on the mainline-modular principle. It includes a parallel processor controller (KPP) and calculating modules organized on synchronous and/or asynchronous principles. The number of modules and their types are determined by the structure of the realized algorithm. A synchronous module is used for the sequential and strongly bound parts of the algorithm and asynchronous modules are used for parallel weakly bound parts. This structure of the processor ensures by the step-by-step nature of fulfilling programs and the problem orientation of the processor.

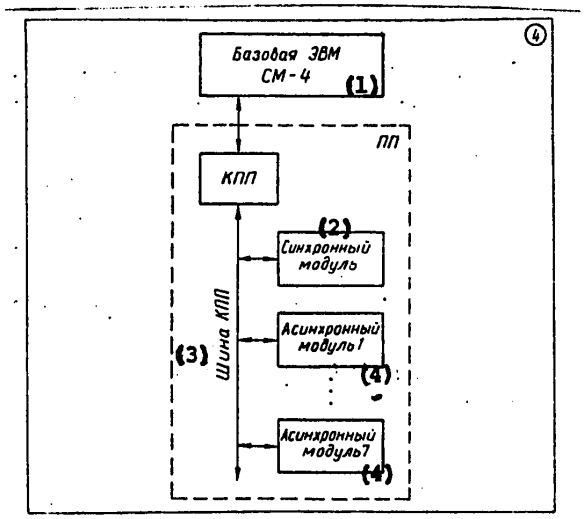


Key:

1. Multiplexer
2. Block for selection and storage of analog signals
3. Comparator
4. Readiness
5. Synchronizer and digital time interval meter
6. Output
7. Synchronization
8. Systems controller
9. Input

The synchronous calculating module (Figure 5) includes an interface module (IM), instruction processor (KP) that includes an instructions memory (PK), microprogram control device (UMU) and microprogram memory (MPP), a set of arithmetic-logic recorder type devices (RALU) with supplementary register storage (SOZU), matrix multiplier block (BUM) and multiblock data memory (PD) with its own controller (KPD). Hybrid type integrators (GI) can also be included in the composition of the module if there is a need to accelerate execution of integration operations.

Data communication between the devices of the module is provided by two parallel-operating data buses ShD1 and ShD2. Control is exercised by control buses ShU and signal of state buses ShS. The synchronous module has the following characteristics in the considered composition: digit capacity of 16-bit word, data memory capacity of 64K words, instruction storage capabity of 1K words, 4 RALU, capacity



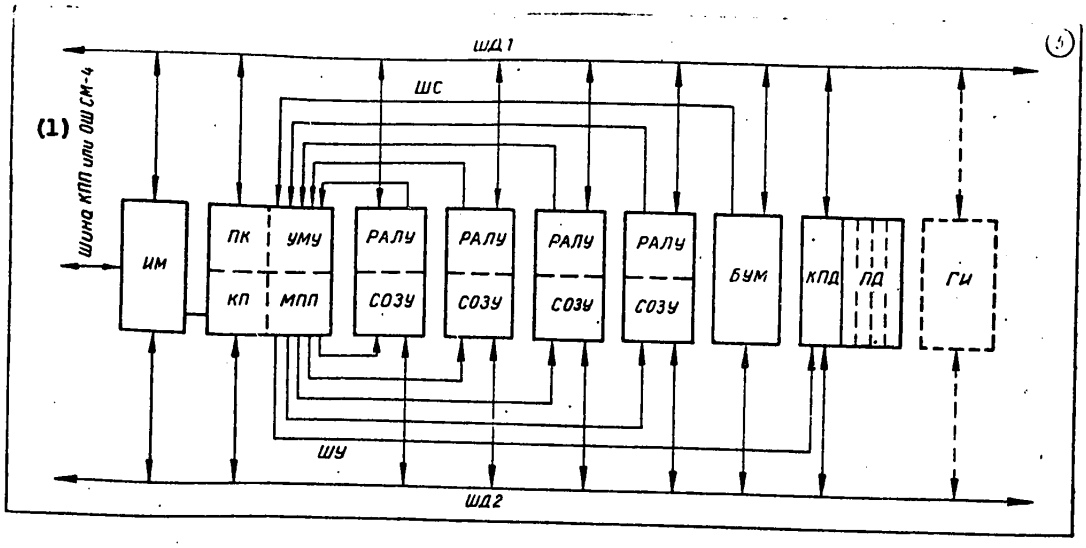
Key:
1. SM-4 baseline computer
2. Synchronous module
3. KPP bus
4. Asynchronous module

of register memory in one RALU is 11 registers, capacity of SOZU in one RALU is 64 words, digit capacity of data buses is 16 bits, width of microinstruction is 128 bits, capacity of microprogram memory is 2K microinstructions, length of a cycle is 250 ns, structure is a section based on 25 cards measuring 210 X 170 mm, consumed power is 300 W and productivity is approximately 100 million equivalent brief operations of the baseline SM-4 computer.

The productivity of the module is evaluated by the standard problem of modelling a flying vehicle with 6 degrees of freedom (system of nonlinear differential equations of 32nd order) with an upper problem frequency of 4 Hz and accuracy of solution not less than 0.01 percent. Productivity is increased 4-6-fold if an additional 30 hybrid integrators (two integrators each on a single standard card) are introduced into the module.

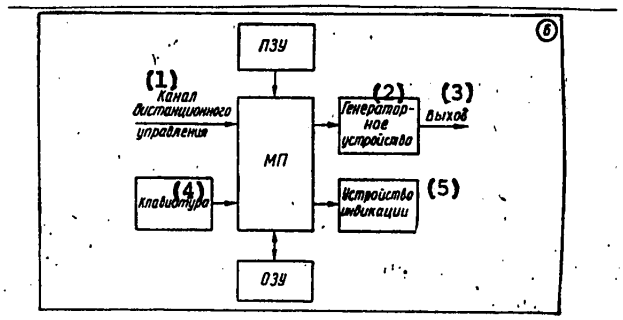
The problem orientation of the structure and instruction system, parallel realization of mathematical operations and control, the use of high-speed bipolar microprocessor sections, matrix multipliers based on BIS and a high-speed semiconductor memory permit a significant increase of the PP speed compared to the baseline SM-4 computer.

Signal generation. A block diagram of the generator of controlled sequences of voltage pulses with built-in microprocessor is presented in Figure 6. Part of the generator that produces the output signals is called the generator device. The microprocessor tunes the generator device and controls its starting and stopping. The main function of the microprocessor is to process the data and instruction flow which come from the keyboard with local control or from the communications



Key:
1. KPP bus or SM-4 common bus

channel with remote control. The numerical values of the parameters available in the data flow should be stored, converted and sent in the form of words of state to the different blocks of the generator. The microprocessor also controls the display in the dynamic mode and realizes the exchange protocol according to the requirements of the interface.



Key:
1. Remote control channel
2. Generator device
3. Output
4. Keyboard
5. Display device

The number of services offered to the operator increases when a microprocessor is used. Because of special subroutines, it becomes possible to check the input values of parameters for permissible values or for conformity of the value of other

parameters with indication of the type of error on an illuminated display board and in emergency situations with the device switched off.

Let us note in conclusion that the microprocessor equipment has a considerable effect on the methods of constructing information, control and calculating systems and the properties of the microprocessor devices permit them to be used effectively for the following purposes:

to automate processes with the simplest algorithms and the use of microprocessor controllers based on the produced MPK provides the highest efficiency in this case;

to automate control processes with adequate complexity of the algorithms for which small computers were previously used;

the use of microcomputers or computer systems is effective in these cases;

to significantly raise the intellectual properties of different monitoring and measuring and control devices by building the microprocessor and microcomputer into the hardware;

to create decentralized information processing systems with structural combination of data processing and gathering devices, which increases the degree of parallelism of the calculating process;

to construct new highly efficient computers and systems of different class.

The main disadvantages that make mass use of microprocessors difficult are the following:

inadequate development of software problems and a lag of investigations to develop debugging devices for the software;

the practical absence of microperipheral devices;

a lag of research in the field of developing problem-oriented microprocessor modules and their software;

the absence of sufficient contact between developers of microprocessors and developers of systems;

inadequate training of personnel to develop microprocessor systems.

BIBLIOGRAPHY

1. Balashov, Ye. P. and G. A. Petrov, "Microprocessors--a New Component Base of Electronics," IZVESTIYA VUZOV, RADIOELEKTRONIKA, Vol 21, No 11, 1978.

2. Balashov, Ye. P., V. B. Smolov, G. A. Petrov and D. V. Puzankov, "Mnogofunktsional'nyye regulyarnyye vychislitel'nyye struktury" [Regular Multifunctional Calculating Structures], Moscow, Sovetskoye radio, 1978.

3. Belous, A. I., M. Yu. Klyashtornyy and S. T. Khvoshch, "Opyt razrabotki spetsializirovannykh mikro-EVM na osnove mikroprotsessora K584 IK1" [Experience of Developing Specialized Microcomputers Based on the IK1 K-584 Microprocessor], Leningrad, LDNTP, 1980.

4. Balashov, Ye. P., V. B. Smolov, D. V. Puzankov et al, "Principles of Designing Numerical Program Control Systems Based on Regular Microprocessor Structures," PRIBORY I SISTEMY UPRAVLENIYA, No 11, 1978

5. Gerasimov, I. V. and S. V. Rodionov, "Integration of Analog-Digital Converter and the IK80 K-580 Microprocessor," PRIBORY I SISTEMY UPRAVLENIYA, No 10, 1980.

6. Balashov, Ye. P., V. Ye. Kochetkov and D. V. Puzankov, "Microprocessor Realization of External Storage Control Devices," PRIBORY I SISTEMY UPRAVLENIYA, No 12, 1978.

7. Mazilkin, I. G., G. A. Petrov and G. Pulkkis, "About the Implementation of Numerical Control of Machine Tools by Multimicroprocessor Systems," HELSINKI UNIVERSITY OF TECHNOLOGY, SERIES B, No 7, 1979.

8. Gordonov, A. Yu., G. A. Petrov, V. S. Stepanov et al, "Specialized Processors in Communications Equipment," VOPROSY RADIOELEKTRONIKI. SERIYA TPS, No 5, 1975.

6521
CSO: 1863/55

UDC 658.012.011.56:681.518

AUTOMATION OF INFORMATION PROCESSES IN INTEGRATED AUTOMATED INDUSTRIAL MANAGEMENT SYSTEMS

Moscow AVTOMATIZATSIYA INFORMATSIONNYKH PROTESSOV V INTEGRIROVANNYKH ASU PROMYSHLENNYMI PREDPRIYATIYAMI in Russian 1981 (signed to press 21 Jul 81) pp 2-3, 105-117, 123-124, 132-133, 141

[Annotation, foreword, sections 7,8,9 (excerpts) and table of contents from book "Automation of Information Processes in Integrated Automated Industrial Management Systems" by Tofik Mamedovich Aliyev, Rafik Azizovich Aliyev and Zinoviy Veniaminovich Khaldey, Energozidat, 6500 copies, 142 pages]

[Excerpts]

Annotation

This book examines questions of automating information processes in integrated management information systems used in industrial enterprises involved in continuous production, as well as the construction of information models, development of information-reference systems, selection of complement of hardware and methods of investigating information flows. Special attention is devoted to experience gained in designing and introducing various information systems within industry.

The book is intended for engineering-technical workers at scientific-research and design organizations involved in developing automated information processing and management systems, as well as students in senior courses at higher educational institutions.

Foreword

Increasing attention has recently been devoted to improving management processes, including those used in industrial enterprises.

The achievements of science and technology make it possible to improve existing management information systems (MIS), and to raise the technical and scientific level of newly created MIS. The greatest development of industrial MIS is reflected in systems which are qualitatively new, namely integrated management information systems. The objective conditions for the creation of such systems consisted of the domestic production of third-generation computers and widely varying types of peripheral devices. In addition, technological processes involved in data acquisition, processing and output have become much more complicated.

One of the most important stages in creating a management information system is the actual technological data processing process. Experience has shown that new management technology is formed on the basis of accumulated experience, and makes extensive use of that experience. The present book disseminates experience in developing information systems in the petroleum processing industry, and devotes attention to the organization of work in the pre-design stage, methods of information research and problems of processing data in integrated management information systems used in petroleum processing enterprises. Practical results associated with these matters obtained at various facilities within the branch are described.

Sections 7, 8, 9 [excerpts]

The use of the Small Computer System (SCS) makes possible a rational combination of centralization and decentralization of technological process monitoring functions, and creates the foundation for using direct digital control and gradually eliminating the use of specialized automation systems and cumbersome instrument panels at the bottom level of automated process control systems [70].

The SM-1 and SM-2 control computer complexes are designed for use in automated equipment, process and production control systems. These complexes are configured at the customer's specifications on the basis of the SM-1P and SM-2P processors using SCS modular assemblies. These can use peripheral devices in the M-6000, M-7000 ASVT-M nomenclature, and are fully compatible with the latter with respect to input-output interface. Interfacing is also supported with the Unified Computer System, KTS LIUS, local information-control system hardware and other systems [71].

The SM-1 and SM-2 control computers can provide the basis for configuring local as well as territorially distributed multi-machine computing complexes.

In order to use the SM-1 and SM-2 control computer complexes in an automated process control system, the users are provided with a set of applications programs which can be used to realize data acquisition, analysis and primary processing functions. The scope of primary processing for analog transducers includes calculation of real values, linearization, smoothing, and introducing temperature and pressure corrections [72].

This hardware makes it possible to synthesize automatic process control system hardware which implements centralized and decentralized structures allowing for the functions which are executed. Figure 15 shows automatic process control system hardware which includes KM2101 and KM2103 local information-control system hardware which can be used to implement a centralized structure in which the KM2103 complex executes the functions of a terminal device for the control computer complex. Figure 16 shows automated process control system hardware built on the basis of SCS which implement a decentralized structure.

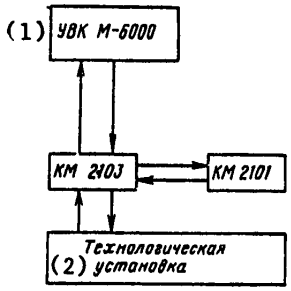


Figure 15. Centralized automated process control system hardware structure

Key: 1. M-6000 control computer complex
2. technological device

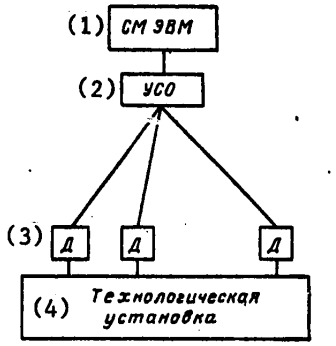


Figure 16. Decentralized automated process control system hardware structure

Key: 1. SCS
2. communication device
3. transducers
4. technological device

The solution of technical and economic problems involved in managing an enterprise as a whole (MIS tasks) is distinguished by the lack of the automatic measurement procedures executed within the group of automated process control system tasks. The other aforementioned functions (recording, conversion, acquisition, transmission, processing, displaying, accumulating and outputting results) also have distinguishing characteristics: information recording consists of entering information in basic documents having an established form and content; data conversion consists of transferring to some medium for computer input; data acquisition, transmission and processing includes storing the source data transmitted from remote facilities to magnetic tapes or discs, manipulating the data during the problem solving process, correcting and forming a series of data files; the output of results includes outputting calculation results for users in hard-copy or video display form.

Data recording and conversion functions are implemented in MIS by means of data preparation devices. The procedure involved in preparing source documents and machine media includes filling out standard blanks manually and then transferring the contents of the source document to machine data media using data preparation devices. Since the process of transferring data to machine media is labor-intensive and involves the possibility of error, a great deal of attention is devoted to improving computer data input techniques. This improvement is moving

in two basic directions: creating automatic readers in which the source document is used for data input; and combining the preparation of the source document and the input of data to the computer in a single process.

Automatic readers and machine-readable documents are not yet being used in the petroleum processing industry integrated MIS; their use is a matter for the future. Implementation of the second direction involves the use of devices which belong to the nomenclature of data teleprocessing system hardware. The use of these devices for the purposes indicated above involves executing the following operations: preparing data for computer input by entering it via keyboard; recording the entered data; editing the data; inputting the data to the computer.

Data teleprocessing equipment includes three groups of devices: data transmission multiplexers, communications devices and subscriber terminals. Data transmission multiplexers are used to interface data transmission equipment to Unified System computers.

There are four types of multiplexers in the Unified Computer System: the MPD-1A, which allows 15 different subscriber terminals to operate independently over communication channels at the same time (the MPD-1 is supplied in two modifications: the basic number of communications channels which can be connected is 32, and can be increased to 64); the MPD-2 provides the capability of connecting between 16 and 176 channels in increments of 8 channels; the fourth multiplexer is the four-channel MPD-3. The communications hardware which comprises the data transmission equipment includes modems, signal converters and error protection devices which are designed to operate over various types of channels. The following types of modems have been developed: modems for operation over telephone and wideband channels at rates of 200-4800 and 48,000 bps, and modems designed to operate over telegraph channels at rates of 50, 100 and 200 bps. The signal converters are designed to operate over physical links at rates of up to 9600 bps. These are produced in several modifications (low-level, telegraph type). Error protection devices are supplied in versions which support half-duplex and duplex working.

Various types of subscriber terminals are used in data processing systems based on the Unified Computer System. The AP-1, AP-2, AP-3 and AP-4 terminals are used for data acquisition. The AP-61, AP-62 and Ap-63 display-based terminals are efficient for inquiries and data processing for inputting and outputting large data files, as well as interactive data processing; the AP-70 and AP-11 are designed for remote data processing. The information from the subscriber terminal is processed by the computer and the results returned to the subscriber terminal [68].

Figure 17 shows a general diagram of data teleprocessing using Unified Computer System facilities for a single communications channel.

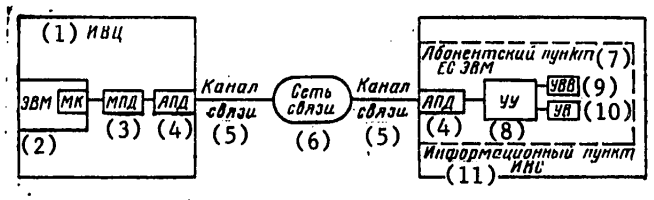


Figure 17. General diagram of data teleprocessing for a single communications channel.

Key: 1. integrated computer center
2. computer (multiplexed channel)
3. data transmission multiplexer
4. data transmission equipment
5. communications channel
6. communications network
7. Unified Computer System subscriber terminal
8. controller
9. data input device
10. data output device

The nomenclature of equipment in the SCS allows them to be used to build data acquisition and transmission systems. The data transmission devices in the SCS include all of the basic modules inherent in large-computer teleprocessing systems: data multiplexers, data transmission devices, modems, signal converters, concentraters and automatic signaling devices.

In addition, the data transmission devices include a number of modules which are aimed especially at SCS use: data transmission adaptors, automatic signaling adapters, programmable error protection devices, simplified modems, signal converters with bi-impulse coding, and modems with simple accuracy enhancement. The use of special-purpose modules as part of the data transmission equipment makes it possible to co-locate SCS with the controlled entity, to use SCS as satellites for large computers, to combine several territorially separated small computers into a multi-machine complex, to simplify procedures for exchanging data with terminal devices and to use a synchronous data transmission [68].

The use of typewriters in Unified Computer System teleprocessing systems makes it possible to combine data input and hard-copy production. However, using typewriters to input data involves a number of shortcomings, which include low input speed, impossibility of editing information prior to input and poor typewriter reliability.

In order to eliminate these shortcomings, displays can be used for data input. However, most hard-copy information is now being input to integrated management information systems by transferring it to machine media centrally within the integrated computer center using appropriate input devices. The following are

provided for Unified System computers: a 1500 cps punch-card reader and a 1000-1500 cps paper tape reader.

The information processing functions are borne by computers in solving management information system problems. The technical system complex in the integrated management information system of the petroleum processing industry uses computers in the Unified System and ASVT series, specifically the YeS-1022 from the Unified System series and the M-4030 from the ASVT series.

Most of the output data resulting from solving management information problems is presented in hard-copy form; therefore, alphanumeric printers are used for data output.

The alphanumeric printers used in the Unified Computer System have a printing rate of 650-900 cps. Data files are output to peripheral devices -- magnetic tapes, discs and, much less often, paper tape and punch cards.

Combined functioning of management information systems and automated process control systems based on using the same basic data poses the problem of creating multi-machine complexes with a broad network of peripheral hardware interconnected via communications channels.

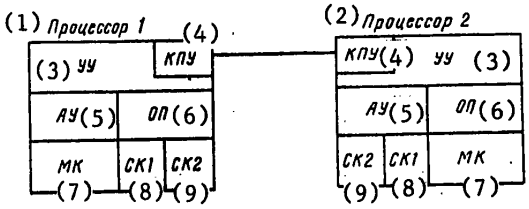


Figure 18. Multi-computer organization at direct control channel level

Key:
1. processor 1
2. processor 2
3. controller
4. direct control channels
5. [probably arithmetic unit]
6. main memory
7. multiplex channel
8. selector channel 1
9. selector channel 2

A multi-machine complex is formed both by combining high-speed computers used to solve management information problems with the control computer complexes making up the automatic process control system hardware, as well as by combining

two or more computers designed to solve management information problems in cases in which the computing capabilities of a single computer are insufficient. The requirement that two computers be used to solve management information problems may also be dictated by reliability considerations. However, it should be kept in mind that the use of two computers to increase productivity results in a mode of distribution of problems between the interconnected computers which is complicated in terms of its organization. It is simpler to organize the functioning of a single powerful computer.

Multi-computer organization can thus occur at the computer level as well as the control computer-computer level in the integrated management information system hardware used in the petroleum processing industry.

Multi-computer organization can be done at four levels [68]: the direct-control channel level, main memory level, computer channel level and peripheral device level.

Multi-computer organization at the direct-control channel level (Figure 18) establishes a connection between the central controllers of the computers and provides highly efficient data exchange between the computers. Multi-computer organization at the main memory level (Figure 19) allows several processors to use a common memory, which provides expanded memory for each processor.

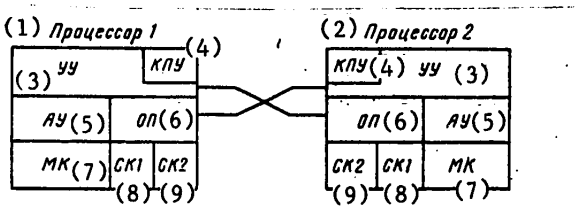


Figure 19. Multi-computer organization at main memory level

Key:
1. processor 1
2. processor 2
3. controller
4. direct control channels
5. [probably arithmetic unit]
6. main memory
7. multiplex channel
8. selector channel 1
9. selector channel 2

When multi-computer operation is organized at the channel level (Figure 20), a connection is established between the computer selector channel, which allows the main memories in the computers to exchange data in a mode in which one of

the computers acts as a peripheral device with respect to another computer. For the Unified Computer System, multi-computer organization at the channel level is done with the help of a "channel-channel adapter". Multi-computer organization at the peripheral device level is done by connecting several computers to the same peripheral device.

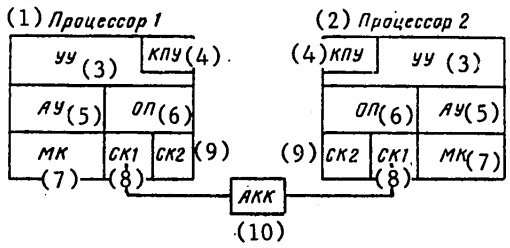


Figure 20. Multi-computer organization at computer channel level

Key:
1. processor 1
2. processor 2
3. controller
4. direct control channels
5. [probably arithmetic unit]
6. main memory
7. multiplex channel
8. selector channel 1
9. selector channel 2
10. channel-channel adapter

The availability of two inputs to the peripheral devices and Unified System multiplexers makes multi-computer organization at this level possible (Figure 21). Multi-computer organization at the peripheral device level makes it possible to connect two computers to the same data transmission multiplexer, which provides uninterrupted operation over communications channels when one of the computers malfunctions.

Unified System computers and M-6000 control computers can be combined to create multi-machine complexes which are not separated by long distances by using channels or an external memory field. The machines are connected by means of an input-output interface matcher and extra program modules which are added to the standard operating system [69]. The matcher can be connected to either a program channel or to the M-6000 direct memory access channel (Figure 22). When connected to a Unified System computer, the matcher is connected to the selector channel as a high-speed peripheral device. Two interface cards are used to connect the matcher: a control interface card and data interface card.

Figure 23 shows methods used for remote interfacing of Small System computer complexes with Unified System computer.

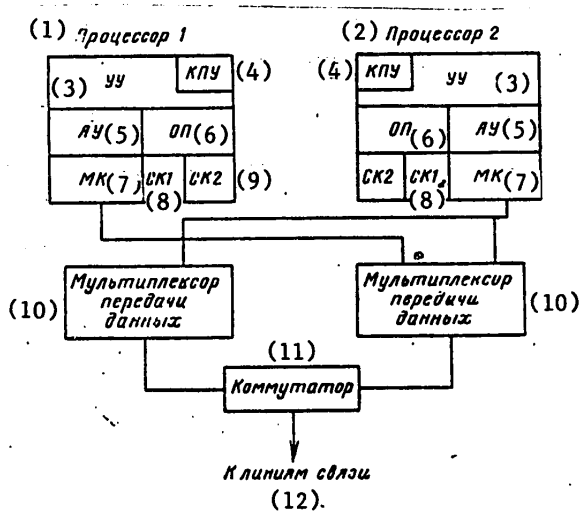


Figure 21. Multi-computer organization at peripheral device level (communications link).

Key:
1. processor 1
2. processor 2
3. controller
4. direct control channels
5. [probably arithmetic unit]
6. main memory
7. multiplex channel
8. selector channel 1
9. selector channel 2
10. data transmission multiplexer
11. switch
12. to communications links

The first two configurations correspond to the interfacing version in which the Small System processors are co-located with the Unified System complex. In this case the Small System computers act as a communications processor which controls the data transmission channels and does preliminary processing of the data passing over them.

The third configuration corresponds to the version in which remote complexes of Small System computers are connected over data transmission channels [73].

The large variety of functional capabilities of broad nomenclature of computer devices make it possible to use Small System computers as measurement-information elements in a measurement-information system. As part of a complex management information system, Small System computers have devices for gathering and processing initial data from different sources; the small computers can also act as a data source for a higher-level computer. Figure 24 shows a structure

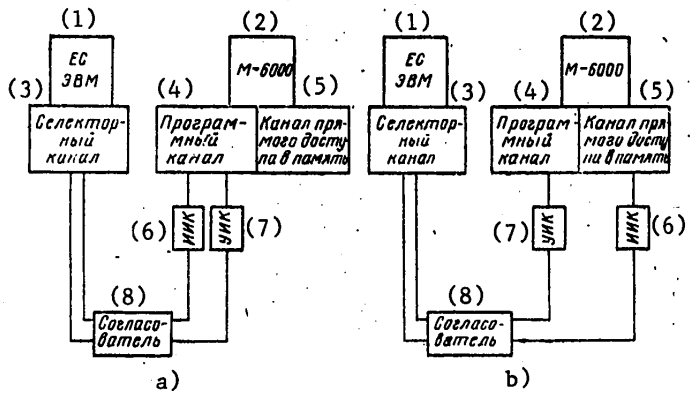


Figure 22. Connecting Unified System computer to M-6000 computer. a -- with matcher connected to M-6000 program channel; b -- with matcher connected to M-6000 direct memory access channel

Key:
1. Unified System computer
2. M-6000
3. selector channel
4. program channel
5. direct memory access channel
6. data interface card
7. control interface card
8. matcher

in which a Small System computer acts as a communications processor working together with a Unified System main processor. Small System multiplexers and adapters are used to interface the processor with the communications channels [74].

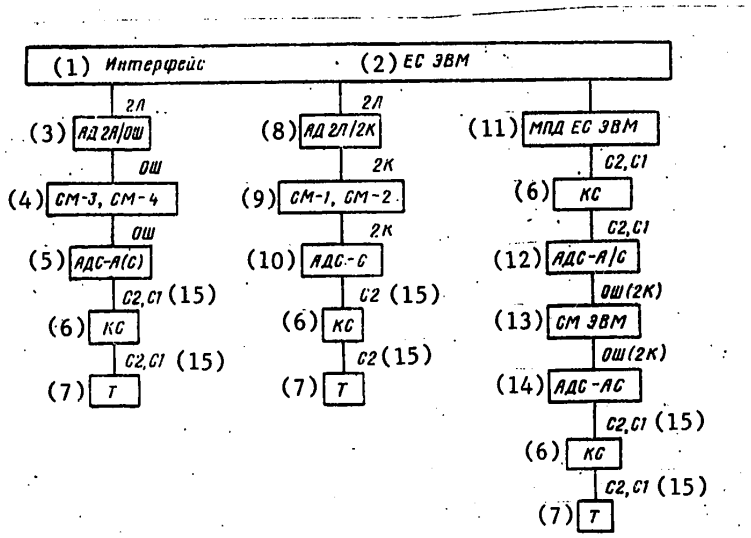


Figure 23. Methods of Small Systems computer interaction with Unified System computers.

Key:
1. interface
2. Unified System computer
3. 2A/OSh rank communications adapter
4. SM-3, SM-4
5. asynchronous (synchronous) data transmission adapter
6. communications channel
7. terminal
8. 2A/2K rank communications adapter
9. SM-1, SM-2
10. asynchronous-synchronous data transmission adapter
11. Unified System multiplexer
12. asynchronous/synchronous data transmission adapter
13. Small System computer
14. asynchronous-synchronous data transmission adapter
15. standard interfaces

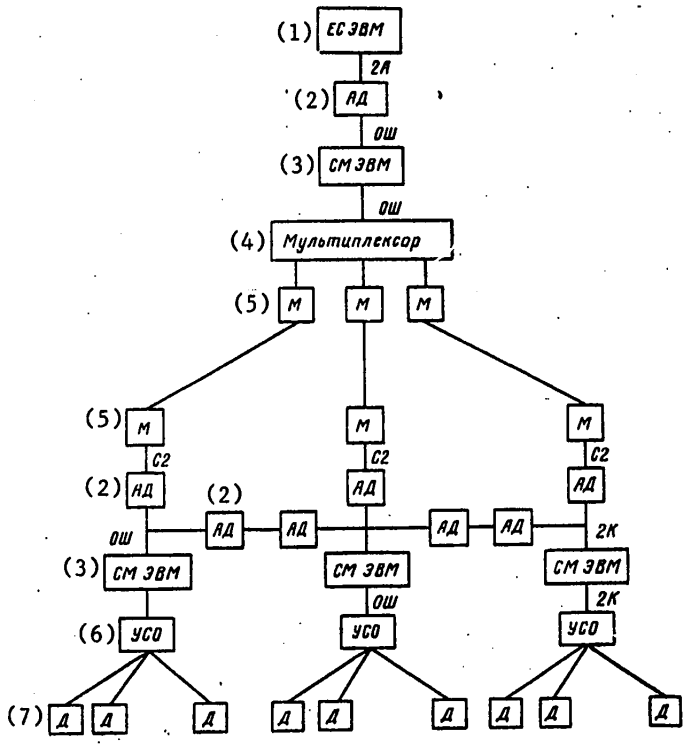


Figure 24. Geographically distributed structure of hardware in technological data acquisition information system

Key:
1. Unified System computer
2. [probably adapter]
3. Small System computer
4. multiplexer
5. [probably modem]
6. computer-measured object communications device
7. transducer

CHAPTER 3

EXPERIENCE IN DESIGNING AND INTRODUCING INFORMATION SYSTEMS

8. Automation of Information Processes in Integrated Management Information System at Novo-Bakinskiy Petroleum Processing Plant imeni Vladimir Il'ich

The Novo-Bakinskiy Order of October Revolution Petroleum Processing Plant imeni Vladimir Il'ich is one of the most modern petroleum processing enterprises in this country. It includes over 20 installations such as the combined high-capacity ELOU-AVT 6 [expansion not given], counterflow-stage catalytic cracking installation, sulfuric-acid alkylation installation, retarded coking installations and others. In order to improve the level of organization of management of the enterprise, the NBNZ imeni Vladimir Il'ich began using scientifically founded mathematical methods to automate management processes at all levels of management of the enterprise [76].

The system was put into full-scale operation in 1975. Effective management of an enterprise as large as the NBNZ required the use of a systems approach to designing the management information system and, consequently, considering it as a multi-level hierarchical organization.

The structure of the management information system for the NBNZ was investigated, and its algorithmic, organizational and functional hierarchies developed. The NBNZ MIS is constructed as a full-level hierarchical system: I -- optimal planning, accounting and analysis of production and management activity; II -- optimal operational control and operational production accounting; III -- optimization of technological installation modes; IV -- regulation of process and equipment parameters.

At level I, problems are solved on the basis of the directive indicators from the Ministry of the Petroleum and Chemical Industry of the Azerbaydzhan SSR and forecasts of material and production resources. Optimal short-term and annual plans are drawn up. Real time monitoring of the course of product output is done at this level.

The execution of optimal plans is taken care of at level II. Based on predictions of the arrival of raw materials, shipment of finished product and the status of technological equipment, models of operational planning and control and operational accounting data for the preceding time period are used to calculate short-term operational production assignments (for a period of 1 day).

Level II is connected to the optimal planning level through a dynamic operational planning model which provides the basis for detailing the production program. Operational control and operational planning are interrelated by the formation of a two-module model of operational production management, where the first module is a model of the plan for the current operating period, and the second module is a model for the remaining intervals of the current operational planning stage.

Level III provides optimal coordination between equipment and processes in complex technological systems; it also ensures optimization of the modes of processes and equipment based on the accepted quality criterion, mathematical models of the processes and systems, operational assignments developed at level II, and current analysis of disturbances.

Equipment parameters are monitored and regulated at level IV. Problems of optimal current and annual planning, optimal planning for technical installation repair schedules, operational control of production, automated accounting of petroleum and petroleum product movement, reserves accounting, dispatcher monitoring and control of technological installations within the plant, optimal control of catalytic cracking processes, and optimal control of the sulfuric acid alkylation installation are now being solved as part of the MIS; other tasks which are being handled include integrated automation (monitoring and regulation) of all technological installations in the plant using high-performance automatic regulation systems implemented on the basis of invariance theory and variable-structure system theory; the cost of goods production, material and technical supply planning, technological resource accounting, movement and realization of finished product accounting, calculations of actual cost and payroll, supplier accounting, daily plant reporting and statistical reporting, as well as a number of other problems. A brief characterization of some of the main tasks of the NBNZ management information system is given below.

The functions of business and operational accounting are implemented by the same devices in accounting for commodity production. The operational accounting system for commodity production at NBNZ imeni Vladimir Il'ich is part of the automated system which accounts for petroleum and petroleum product movement. The operational accounting information-computing system for commodity production is a group acquisition and processing system for primary measurement information, and has a two-stage hierarchical structure (Figure 25).

The first stage of this system is implemented using a "Radius" information-measurement system [77]. The second stage includes an M-6000 control computer system, which bears the functions of computational operations as well as centralized selection of the appropriate first-stage information-measurement system. The first-stage information-measurement system includes the monitored facilities, which are reservoirs in which slit transducers are installed along with equipment for piezometric measurement of the hydrostatic pressure of the produce within a group of reservoirs.

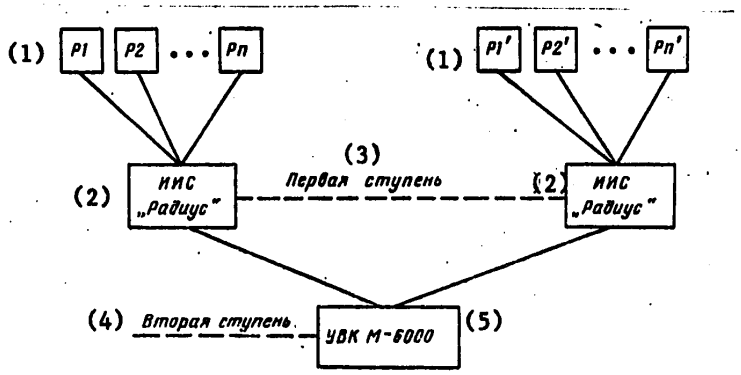


Figure 25. Information-computing system for operational commodity production accounting

Key:
1. reservoirs
2. "Radius" information-measurement system
3. first stage
4. second stage
5. M-6000 control computer system

9. Operational Information Processing System for Petroleum Processing Enterprise Complex

The information system within the automatic management system of the enterprises of the Ministry of the Petroleum and Chemical Industry of the Azerbaydzhan SSR, which operates as part of the unified management information system of that Ministry, is designed to reflect in a timely manner the execution of the production program of the group of petroleum processing plants and enterprises which makes up the Ministry of the Petroleum and Chemical Industry of the Azerbaydzhan SSR.

The technical base of the system is the junction computer center, which includes a YeS-1020 computer and data transmission facilities located at Ministry enterprises. The tasks carried out by the information system include the "Daily Report of the Azerbaydzhan SSR Ministry of the Petroleum and Chemical Industry on the course of execution of the production program", "Operational Report on Functioning of Enterprises of Azerbaydzhan SSR Ministry of the Petroleum and Chemical Industries" and "Report on Operation of Petroleum Processing Plant Group and Enterprises of Azerbaydzhan SSR Ministry of the Petroleum and Chemical Industries" (monthly cross-section).

The daily report presents a picture of the execution of the plan by enterprises belonging to the Azerbaydzhan SSR Ministry for the current day and since the beginning of the month. The operational report on functioning of Azerbaydzhan SSR Ministry enterprises is formed on the basis of data obtained by solving the problem of compiling the daily report, and reflects the course of plan fulfillment for the reporting week and from the beginning of the month. The report on the work of the petroleum processing plant group and enterprises of the Azerbaydzhan SSR Ministry of the Petroleum and Chemical Industry includes summary data on the results of operation in terms of the entire nomenclature of petroleum products produced during the past month. The function of this information system is thus to obtain reliably timely information by means of automated data processing.

Table of Contents

Foreword 3
Introduction 4

Chapter 1. DESIGNING INFORMATION SYSTEMS FOR INDUSTRIAL ENTERPRISES WITH CONTINUOUS PRODUCTION 10

1. Problems of Management and Data Processing at Enterprises with Continuous Production 10
2. Planning Set of Problems in Petroleum Processing Industry Integrated Management Information System 15
3. Information Investigation of Enterprise in Order to Create Integrated Management Information System 28
4. Investigation of Information-Logical Interconnection Between Tasks Involved in Managing Enterprises with Continuous Production 42

Chapter 2. AUTOMATION OF DATA PROCESSING

5. Acquisition, Processing and Representation of Data in Petroleum Processing Industry Integrated Management Information System 69

6. Data Base in Integrated Management Information System of Enterprises with Continuous Production 78
7. Technical Equipment for Automating Information Processes 97

Chapter 3. EXPERIENCE IN DESIGNING AND INTRODUCING INFORMATION SYSTEMS 115

8. Automation of Information Processes in Integrated Management Information System at Novo-Bakinskiy Petroleum Processing Plant imeni Vladimir Il'ich 115
9. Operational Information Processing System for Petroleum Processing Enterprise Complex 132

Bibliography 136

6900
CSO: 1863/58

ORGANIZATIONS AND PERSONALITIES

UDC 518.1

MOSCOW UNIVERSITY'S COMPUTER CENTER

Moscow VESTNIK MOSKOVSKOGO UNIVERSITETA, SERIYA 15: VYCHISLITEL'NAYA MATEMATIKA I KIBERNETIKA in Russian No 4, Oct-Dec 81 (manuscript received 6 Jul 81) pp 3-8

[Article by Ye. A. Grebenikov and V. I. Dmitriyev: "On the Development of Scientific Trends in Moscow University's Computer Center"]

[Text] The Scientific-Research Computer Center (NIVTs) was founded at Moscow University in 1955 in the Division of Mechanics and Mathematics. Scientific management of the computer center's work was conducted by the Department of Computational Mathematics. The scientific work of the center and of this department were closely connected. Theoretical investigations were performed on numerical methods of solving mathematical problems and solutions were found to diverse problems originating in the departments of MGU (Moscow State University) and non-university organizations. The Computer Center was equipped with the first Soviet computer "Strela" and had a relatively small number of workers.

The development of computational mathematics and computer technology, the growing demands of the subdivisions of Moscow University for the use of computers for scientific investigations and in the instructional process and the necessity for increasing the number of trained specialists in applied mathematics stimulated the development of the Computer Center. In 1958, academician A. N. Tikhonov became head of the Department of Computational Mathematics and the scientific director of the Computer Center. Under his influence and with his direct participation, the major scientific trends to be followed by the Computer Center took form and the Center quickly grew into a substantial scientific establishment, equipped with contemporary computer technology and employing a large number of highly qualified specialists. In 1972, the Computer Center was reorganized into the Scientific-Research Computer Center of MGU, a part of the Division of Computational Mathematics and Cybernetics. The breadth of its range and the profundity of the fundamental and applied investigations it has conducted, make it possible to include the NIVTs in the ranks of the leading institutes in our nation in the field of computational mathematics.

The major tasks of the NIVTs at the present time are:

- the development of theoretical problems and applied questions in contemporary computational mathematics which are important for the development of science and the economy;

- participation in training and retraining of scientific cadres in the area of computer utilization, provision of computer management of the instructional process and organization of practical training for students;

- provision of aid in the incorporation and organization of computer utilization for scientific investigations performed by subdivisions of MGU.

The scientific investigations performed at the NIVTs gradually began to center around two leading problems. The first is associated with the development of methods of mathematical modeling, numerical analysis and automated processing of the results of observations. Relevant to the second problem, investigations and development of software and hardware devices are conducted in order to facilitate the effective use of the computer in automated scientific investigations, the instructional process and administration. Within the framework of these two problems, developments are being conducted in accordance with many scientific trends. The establishment and further development of the majority of these scientific trends is closely associated with the name of A. N. Tikhonov. Of the seven doctoral dissertations defended by workers of the NIVTs, six were defended by A. N. Tikhonov's students. A. N. Tikhonov's work is characterized by the juxtaposition of basic mathematical problems and work on actual natural science topics. This has put a distinctive stamp on the work of the NIVTs as well, where, along with basic research on computational mathematics, work is conducted, along a broad front, on the construction of mathematical models and the numerical analysis of them for many important economic tasks.

In the Scientific-Research Computer Center, in cooperation with the Department of Computational Mathematics of the MGU Division, they are currently conducting wide scale investigations to develop principles and methods of mathematical modeling of the phenomena and processes involved in various natural sciences and to produce complex engineering designs. The goal of this work is an increase in the efficacy of the use of computers in automated scientific investigations, the development of methods of automated processing and interpretation of experimental data and the creation of systems to search for optimal designs in the solution of scientific engineering problems. Within the framework of this problem, they are conducting research to create mathematical models of electrodynamics, geophysics, plasma physics and aeromechanics. They are also developing numerical methods for solving problems in mathematical physics and standard problems in numerical analysis. A large cycle of work is related to the creation of methods for solving inverse problems and, based on this, the development of systems of processing and interpretation of experimental data. Let us briefly consider the development of these scientific trends.

Computational electrodynamics began to develop in the NIVTs starting in 1958, when, in response to an initiative by A. N. Tikhonov, a small scientific group was created. In this group they developed numerical methods for the solution of boundary-value problems in electrodynamics, applied to problems in geoelectricity, diffraction of electromagnetic waves and the theory of irregular waveguides. Research in this direction developed very rapidly and was widely acknowledged in our nation, as well as abroad. At the same time, work was performed on the adoption of computers in the area of construction of antenna feeder devices, synthesis of new types of antenna and the quantitative description of propagation of radio waves in non-homogeneous media.

Principles were developed for a new approach to tasks in design of complex radiating systems, based on the general theory of regularization of solutions to poorly defined problems in mathematical physics. As a result of the research performed, a method of solution was developed for the task of symthesizing antenna systems with given requirements for the radiation direction diagram, taking account of actual constraints on the arrangement of sources of excitation, such as arise in practical problems of antenna design. The adoption of the methods developed in the practice of design of antenna devices made it possible to improve sustantially the parameters involved in the construction of radiating systems for various purposes. For this cycle of work, academician A. N. Tikhonov and professors V. I. Dmitriyev, A. S. Il'inskiy and A. G. Sveshnikov were granted the title of winners of the USSR State Prize.

Effective methods for performing calculations in solving problems related to the theory of regular waveguides were developed, based on the general ideas of the Bubnov-Galerkin method, which reduced problems in the propagation of electromagnetic waves in waveguides to the solution of boundary value problems for a finite system of ordinary differential equations; normal waves in corrugated waveguides and open micro-planar transmission lines were also studied and a method for calculating the continuous propagation of these waves was developed. On the basis of the realization of the algorithms developed, research was performed on specific directing systems.

Very important work was performed on the creation of numerical methods for solving problems in diffraction by a local non-homogeneous body and the development of methods of numerical analysis of periodic antenna grids. Also important is work investigating problems of diffraction by ideally conductive screens. This cycle of work has great significance for the creation of modern antenna systems since it makes it possible to create mathematical models of radiating systems with various structures.

Within the framework of the development of numerical methods in electrodynamics, work was started on geoelectricity. Here, under the direction of A. N. Tikhonov, research was performed on mathematical models for problems arising in the theory and interpretation of geophysical methods of prospecting for commercial minerals.

These methods are based on the use of artificially and naturally created electrodynamic fields. Later, this work trend split off to become an independent effort and was expanded significantly.

The major goal of all geophysical research is the solution of inverse problems, i.e., determination of the structure of a medium form measurements of the characteristics of a field. An electromagnetic field, measured at the surface of the Earth, carries information about the electrical structure of the Earth. To isolate this information, one must solve an inverse problem. This can be done on the basis of the repeated solution of the direct problem for a selected class of models of the structure of the medium. For this reason, the creation of numerical methods for solving direct problems involving electromagnetic soundings of the medium, in many respects, determines whether it is possible to introduce various geophysical methods into the actual search for commercial minerals. Thus, A. N. Tikhonov's method for calculating the electromagnetic characteristics of fields in stratified media made it possible to begin adopting the method of frequency sounding in the economy. This facilitated the technical re-equipment of electro-prospecting for commercial minerals.

A. N. Tikhonov proposed a method of magnetoterrestrial sounding of the Earth's crust and upper mantle based on study of simultaneous measurements at the Earth's surface of the tangential components of the natural magnetic field.

A cycle of work was performed at the NIVTs involving numerical modeling of the magnetoterrestrial sounding of the Earth's crust and upper mantle, based on the study of simultaneously measured tangential components of the natural magnetic field at the Earth's surface and of non-homogeneous media. Methods were developed for solving inverse problems related to magnetoterrestrial sounding based on the regularization method.

A cycle of work was performed at the NIVTs on methods of electroprospecting for ore. Methods were developed, which were based on numerical calculation of models of local conductive inclusions in a stratified medium, for the interpretation of data from electro-prospecting for ore. Important work was performed on mathematical modeling of the tasks of electrical well logging. Methods were also developed here which make it possible to solve problems of axiosymmetrical distribution of electrical conductivity, when parameters of the medium change as a function of depth, as well as radius. This makes it possible to numerically study mathematical models which adequately represent actual geological situations.

A library of programs "Electromagnetic fields in geophysics" was created containing programs for solving direct and inverse problems related to electromagnetic sounding. The most effective and general method of approximate solution of differential equations is the method of finite differences. At the end of the 50's, A. N. Tikhonov and A. A. Samarskiy performed some fine-grained investigations of a broad class of difference diagrams for the solution of

boundary value problems for differential equations with smooth and discontinuous coefficients (homogeneous difference diagrams). Applying theoretical-functional methods to the study of difference diagrams, they obtained important results on the convergence and stability of difference diagrams for the class of equations with discontinuous coefficients. In the Computer Center, a group was created which, under the leadership of Academician A. A. Samarskiy, began work on the theory of difference diagrams and its applications. In the past few years, this scientific trend has expanded strongly and has encompassed research in the development of net approximations of boundary value and initial boundary value problems for elliptical, parabolic and hyperbolic equations, the study of stability and convergence of difference diagrams and the development of methods for solving net equations. Work was conducted in the application of the methods developed to problems in plasma physics and the theory of elasticity.

A cycle of work was performed in the NIVTs on the study of difference diagrams used in the search for visual solutions to differential equations. A class of diagrams was identified for equations of thermal conductivity, which were most suited to the solution of problems with discontinuous initial conditions. The precision of difference diagrams was studied for second order elliptical equations in the presence of a first order discontinuity in the Dirichlet boundary conditions. The convergence of difference diagrams was studied for Laplace equations at the angles and modified diagrams with more rapid convergence were constructed. The features of difference diagrams were studied as applied to the crack problem, the aperture problem and a number of other problems.

Effective methods for solving net equations were developed including: a) an iterative method involving variable directions of a high level of accuracy for solving Poisson equations and the third boundary value problem for elliptical equations in a rectangle; b) a modified variable-triangular method which permitted effective solution of elliptical equations with strongly varying coefficients; c) a method using hypothetical unknowns for solving Dirichlet problems for the case of a Poisson equation in an irregular region. A substantial contribution was made to the development of the general theory of iterative methods; in particular, the stability of a two layer Chebyshev iterative method was studied and algorithms for selecting an iterative parameter were proposed.

These methods and algorithms formed the basis for the development of a series of effective programs. A set of programs was created to solve linear and quasilinear equations of thermal conductivity, problems of the Stefan type and equations from the theory of elasticity. The development of a set of programs to solve boundary value problems for elliptical equations has been completed. The methods developed are used actively in the study of various nonlinear problems in plasma physics. Research is being conducted on the processes of shaping, focussing and transporting intense electron and ion beams, their interaction with plasma and also on the study of questions related to the stability of a plasma column. Numerical methods for studying processes of plasma flow in a magneto-plasma compressor are being developed.

A cycle of research is being conducted to develop numerical methods for solving applied problems in aeromechanics. This scientific trend was organized in the NIVTs by Academician G. I. Petrov. Under his direction fundamental investigations were carried out on the solution of applied problems in aerodynamics. Methods for numerical calculation of flow in wind tunnels were developed. These provided the basis for analyzing spatial flows in supersonic wind tunnels, studying the influence on the structure of flow of spatial effects arising as a result of irregularities of the stream when it enters the wind tunnel or of disruption of the symmetry of the walls of the wind tunnel. The streamline flow around sharp, multistage cones was studied numerically, as was streamline flow of an ideal gas and stream of stable air around the spherical surface of blunted cones and blunted wedges. A significant cycle of work was performed in calculating the flows of a viscous thermal conducting gas on the basis of complete Navier-Stokes equations. A detailed numerical study was performed of the structure of a viscous shock layer during super and hypersonic laminar flow around blunted bodies. A package of applied programs for scientific investigations in the field of aerodynamics has been created.

Another scientific trend in the NIVTs involves the development of methods for solving incorrectly stated problems and for automated processing and interpretation of observations. The establishment and development of this scientific trend became possible in the early 60's, after A. N. Tikhonov developed the theory of regularization of incorrectly stated problems. On the basis of a single methodology using a regularizing algorithm for solving inverse problems in mathematical physics, effective numerical methods for processing and interpreting the results of various experiments in natural science were developed. The theoretical results were incorporated in the form of a specialized library of programs for the processing and interpretation of the results of various spectrometric experiments, structural analysis, Mössbauer spectroscopy, and for the solution of inverse problems in geophysics.

Under the direction of A. N. Tikhonov, work was performed to create a theory for complete mathematical processing of the results of observations and multipurpose systems for these problems. Algorithms for solving various inverse problems during the interpretation stage were created. These were stable with respect to disturbances in experimental information. Various stochastic mathematical models were developed in experimental investigations. Methods for statistical processing and interpretation of experimental results were created and implemented. This trend was made possible by the fact that the subdivisions of MGU had adopted computer technology and modern methods of information processing for use in scientific research. The theory and application of mathematical modeling in processing and interpreting experimental data is being developed in cooperation with many laboratories of NIIYaF (Scientific Research Institute of Nuclear Physics), a number of departments of the Division of Physics and also with certain subdivisions of the Divisions of Chemistry and Geology of Moscow University. The creation of the necessary software components and the implementation of a number of automated

systems of data processing and interpretation have made it possible to utilize computer processing of experimental results in the work of the subdivisions of the university. Research was done on the construction of mathematical models of Mössbauer spectra and methods of processing them. Direct and inverse problems involving calculation of spectra of hyperfine combined interactions were posed and solved. Algorithms for figuring out the influence of apparatus factors on experimental spectra were developed. A cycle of work was performed aimed at constructing mathematical models of physics experiments involving the study of the composition of substances and the structure of crystals and molecules using methods of diffractive Roentgen, electronic and neutron analyses. Systems for processing data from photonuclear interactions were created.

Based on the work described, it was possible to create a system for automating scientific investigations with the goal of increasing their efficiency and speeding up the processing of experimental data. In recent years at Moscow University, the automation of scientific research has involved a broad range of activities--from the construction of state-of-the-art mathematical models of natural sciences and of methods of optimal design, control and processing of the results of observations, to the realization of specific measurement-computational units and the creation of problem oriented software. All this necessitated the creation of a collective computer use system to facilitate user access to computer capacities and increase the efficiency of use of computer technology.

The creation of a system of collective computer use for MGU was important not only for automation of scientific research. It was also essential for improvement of the instructional process and administration of Moscow University. At the beginning of the 70's, at the initiative of A. N. Tikhonov, work was begun at NIVTs to create an information system for administration of MGU and also to develop and adopt computer and software aids to permit effective use of the computer in the process of instruction. A system of specialized software was created which made it possible to increase the level of use of the computer in a computer practicum. The development of automated information systems for various purposes was begun. Work toward raising the level of computer services offered to users was intensified. A library of programs on numerical analysis was developed and the possibilities for terminal access to the computer were expanded. All this work formed the basis for creation of a collective use computer system for MGU in the 10th Five-Year Plan. This work was focussed on the leading problem of the creation of a central computer complex for MGU based on the NIVTs computer and of developments in automated scientific research, use of the computer in the instructional process and administration of MGU, the creation of multimachine complexes and a broad terminal net.

The conducting of research and development toward the creation of a Computer Center with a system of collective computer use will facilitate broader adoption of computers and an increase in efficiency of their use in subdivisions of Moscow University.

9285
CSO: 1863/71

FIFTIETH BIRTHDAY OF KAMIL' AKHMETOVICH VALIYEV

Moscow MIKROELEKTRONIKA in Russian Vol 10, No 4, Jul-Aug 81 p 291

[Editorial: "On the 50th Birthday of K. A. Valiyev]

[Text] Corresponding member of the USSR Academy of Sciences Kamil' Akhmetovich Valiyev has celebrated his 50th birthday.

K. A. Valiyev was born in Verkhniy Shander village, Takanyshskiy Rayon, Tatarskaya ASSR in a collective farm worker's family. After graduating from secondary school, he studied at the Physicomathematical Faculty, Kazan' State University imeni V. I. Lenin, graduated from the university with excellent marks and was taken into post-graduate study in the speciality "theoretical physics." Upon completing his graduate study, he was assigned to work at Kazan' Pedigogical Institute, where he worked from 1957 through 1964 in the position of head of the chair of physics. During this period K. A. Valiyev carried out an extensive cycle of theoretical research on Brownian rotation of molecules in liquids and magnetic phenomena in crystals and liquids using methods of nuclear and electron magnetic resonance, infrared spectroscopy and Raman spectroscopy. The results achieved by Valiyev are widely known to specialists and are now being developed actively in the scientific school which he founded at Kazan'.

During the past few years Valiyev's name has been closely related to development of microelectronics technology and development of integrated circuits in our country.

He now heads a sector of the Physics Institute of the USSR Academy of Sciences imeni P. N. Lebedev. His range of scientific insterests includes scientific problems of cathode-ray, ion beam and X-ray methods of lithography and high-speed microelectronic components.

K. A. Valiyev is a doctor of physicomathematical sciences (since 1963), a professor (since 1966) and a corresponding member of the USSR Academy of Sciences (since 1972 in the specialty "engineering physics"). He was awarded the Lenin Prize in 1974 in the field of science and technology and was awarded the State Prize, Azerbaijan SSR in 1976.

K. A. Valiyev, a professor at the Moscow Institute of Electronic Technology, a member of the plenum and council on electronics, USSR high degree commission,

office of the general physics and astronomy department, USSR Academy of Sciences, a number of scientific councils and the editorial board of MIKROELEKTRONIKA, USSR Academy of Sciences, carries out active scientific organizational and pedagogical work, has trained more than 30 candidates and doctors of sciences and works in the scientific personnel certification system. K. A. Valiyev is the author of more than 150 scientific papers, inventions and several monographs.

K. A. Valiyev, a member of the CPSU since 1954, participates actively in social life. He has repeatedly been elected a deputy to the regional council of working people's deputies, a member of the Plenum of the CPSU Raykom and the institute's party committee.

K. A. Valiyev's services have been noted by high state awards--two Orders of the Red Banner of Labor and medals.

We wish Kamil Akhmetovich on his 50th birthday strong health and great new success in his fruitful scientific and scientific organizing activity.

6521
CSO: 1863/54

- END -